DETECTION OF MICROSLEEPS FROM THE EEG VIA OPTIMIZED CLASSIFICATION TECHNIQUES

JOHN LAROCCO

A thesis submitted for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

at the

University of Canterbury,

Christchurch, New Zealand.

13 October 2015

“The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.”

–Howard Phillips Lovecraft

ABSTRACT

Microsleeps are complete breaks in responsiveness for 0.5–15 s. They can lead to multiple fatalities in certain occupational fields (e.g., transportation and military) due to the need in such occupations for extended and continuous vigilance. Therefore, an automated microsleep detection system may assist in the reduction of poor job performance and occupational fatalities. An EEG-based microsleep detector offers advantages over a video-based microsleep detector, including speed and temporal resolution. A series of software modules were implemented to examine different feature sets to determine the optimal circumstances for automated EEG-based microsleep detection.

The microsleep detection system was organized in a similar manner to an EEG-based brain-computer interface (BCI). EEG data underwent baseline removal and filtering to remove overhead noise. Following this, feature extraction generated spectral features based upon an estimate of the power spectrum or its logarithmic transform. Following this, feature selection/reduction (FS/R) was used to select the most relevant information across all the spectral features. A trained classifier was then tested on data from a subject it had not seen before. In certain cases, an ensemble of classifiers was used instead of a single classifier. The performance measures from all cases were then averaged together in leave-one-out cross-validation (LOOCV).

Sets of artificial data were generated to test a prototype EEG-based microsleep detection system, consisting of a combination of EEG and 2-s bursts of 15 Hz sinusoids of varied signal-to-noise ratios (SNRs) ranging from 16 down to 0.03. The balance between events and non-events was varied between evenly balanced and highly imbalanced (e.g., events occurring only 2% of the time). Features were spectral estimates of various EEG bands (e.g., alpha band power) or ratios between them. A total of 34 features for each of the 16 channels yielded a total of 544 features. Five minutes of EEG from eight subjects were used in the generation of the dummy data, and each subject yielded a matrix of 300 observations of 544 features.

Datasets from two prior microsleep studies were employed after validating the system on the artificial data. The first, Study A (N = 8), had 16 channels sampled at 256 Hz from two 1-hour sessions per subject and the second, Study C (N = 10), had one 50-min session with 30-62 channels per subject sampled at 250 Hz. A vector of 34 spectral features from each channel was concatenated into a feature vector for each 2-s interval, with each interval having

a 1-s overlap with the prior one. In both cases, microsleeps had been identified via a combination of video recording and performance on a continuous tracking task.

Study A provided four datasets to compare effects of various preprocessing techniques on performance: (1) Study A bipolar EEG with Independent Component Analysis (ICA) preprocessing and artefact pruning (total automated rejection of artefact-containing epochs) and logarithmic transforms of the spectral features (SABIL); (2) Study A bipolar EEG with ICA-based eye blink removal and artefact removal with pruning of epochs with major artefacts, and linear spectral features (SABIS); (3) Study A referential EEG unprocessed by ICA with spectral features (SARUS); and (4) Study A bipolar EEG unprocessed by ICA with spectral features (SABUS). The second study had one primary feature set, the Study C referential EEG ICA preprocessed spectral feature (SCRIS) variant. LOOCV was evaluated based on the phi correlation coefficient.

After replicating prior work, several FS/R and classifier structures were investigated with both the artificially balanced and unbalanced data. Feature selection/reduction methods included principal component analysis (PCA), common spatial patterns (CSP), projection to latent structures (PLS), a new method based on average distance between events and non-events (ADEN), ADEN normalized with a *z*-score transform (ADENZ), genetic algorithms in concert with ADEN (GADEN), and genetic algorithms in concert with ADENZ (GADENZ). Several pattern recognition algorithms were investigated: linear discriminant analysis (LDA), radial basis functions (RBFs), and Support Vector Machines with Gaussian (SVMG) and polynomial (SVMP) kernels. Classifier structures examined included single classifiers, bagging, boosting, stacking, and adaptive boosting (AdaBoost).

The highest LOOCV results on artificial data (SNR = 0.3) corresponded to GADEN with 10 features and a single LDA classifier with a mean phi value of 0.96. Of the four Study A datasets, PCA with 150 features and a stacking ensemble achieved the highest mean phi of 0.40 with the SABIL feature set, and ADEN with 20 features with a single LDA classifier achieved the highest mean phi of 0.10 with Study C.

Other machine-learning methodologies, such as training on artificially balanced data, decreasing the training size, within-subject training and testing, and randomly mixed data from across subjects, were also examined. Training on artificially balanced data did not improve performance. An issue found by performing within-subject training and testing was that, for certain subjects, a classifier trained on one-half of the subject's data and then tested on the other half was that classifier performance dropped to random guessing.

The low phi values on within-subject tests occurred independently of the feature selection/reduction method explored. As such, performance of a standard LOOCV was often dependent on whether a particular testing subject had a low (< 0.15) within-subjects mean phi correlation coefficient. Training on only the higher mean phi values did not boost performance. Additional tests found correlations (r = 0.57, p = 0.003 for Study A and r = 0.67, p < 0.001 for Study C) between higher within-subject mean phi values (> 0.15) and longer mean microsleep durations. Other individual subject characteristics, such as number of microsleeps and subject age, did not have significant differences.

The primary findings highlighted the strengths and limitations of supervised feature selection and linear classifiers trained upon highly variable between-subject features across two studies. Findings suggested that a classifier performs best when individuals have high mean microsleep durations. On the configurations investigated, preprocessing factors, such as ICA preprocessing, feature extraction method, and artefact pruning, affected the performance more than changing specific module configurations.

No significant differences between the SABIL features and the lower performing Study A feature sets were found due to overlapping ranges of performance (p = 0.15). The findings suggest that the investigated techniques plateaued in performance on the Study A data, reaching a point of diminishing returns without fundamentally changing the nature of the classification problem. The different number of channels of varying quality across all subjects in Study C rendered microsleep classification extremely difficult, but even a linear classifier can properly generalize if exposed to a large enough variety of data from across the entire set. Many of the techniques explored are also relevant to other fields, such as brain-computer interface (BCI) and machine learning.

ACKNOWLEDGEMENTS

I would like to thank all of those who assisted with this research and provided the valuable feedback and resources to see it concluded.

I would like to thank supervisors, first and foremost, for their patience and insight into this project, and for providing me with the opportunity to contribute towards it. Professor Richard Jones, Professor Phil Bones, Dr. Steve Weddell, and Dr. Carrie Innes all were invaluable guides. Professor Richard Jones provided vital expertise on my work. Doctor Carrie Innes was key contributor on a variety of topics. Professor Phil Bones assisted, despite his responsibilities as the head of the Department of Electrical and Computer Engineering. Dr. Weddell lent his own experience to cure my own nescience.

I would like to thank the organizations that made this research possible: the University of Canterbury (UC), the New Zealand Brain Research Institute (NZBRI), Christchurch Neurotechnology Research Programme (NeuroTech), and Canterbury Medical Research Foundation (CMRF). UC provided the opportunity to embark on this exciting research. The NZBRI and the amazing people there provided plenty of food for thought. NeuroTech provided the scholarship that enabled me to pursue this fascinating research. The CMRF provided the travel grant that allowed me to present at the 2014 IEEE Engineering in Biology and Society (EMBS) conference in Chicago.

I would also like to thank my parents for their patience and in helping me throughout my education. In addition, I would also like to thank all of my friends for their continued support.

TABLE OF CONTENTS

PREFACE

This research was completed between May 2012 and May 2015 while enrolled at the University of Canterbury Department of Electrical and Computer Engineering. The research was carried out with NeuroTech at the NZBRI. I was supervised by Professor Richard Jones, Professor Phil Bones, Dr. Steve Weddell, and Dr. Carrie Innes.

PUBLICATIONS

Conference Papers

- LaRocco, J., Innes, C.R.H., Bones, P.J., Weddell, S., & Jones, R.D. (2014). Optimal EEG feature selection from average distance between events and non-events *Proceedings of Annual International Conference on Engineering in Medicine and Biology Society, 36*, 2641-2644.

Published Abstracts

- LaRocco J, Innes C, Bones P, Weddell S, Jones R (Nov 2014). Optimal EEG feature selection from average distances between events and non-events. (Abstract) *Australasian Physical & Engineering Sciences in Medicine*, 38: 195-196. [Presented at New Zealand Physics and Engineering in Medicine Conference (NZPEM 2014), Christchurch, Nov 2014]

Presentations

- August 2014, "Optimal EEG feature selection from average distance between events and non-events," IEEE EMBS Conference. Poster presentation.
- August 2012, "Automated detection and classification of lapses," NZBRI seminar. Oral presentation.

LIST OF ABBREVIATIONS

General Terminology:

LOOCV: “Leave-one-out” cross-validation

MS: Microsleep

WS: Within-Subject

Feature Selection/Reduction (FS/R) Modules:

PCA: Principle Component Analysis

CSP: Common Spatial Pattern

ICA: Independent Component Analysis

ADEN: Average Distance between Events and Non-Events

$ADEN_{10}$: Average Distance between Events and Non-Events, with 10 features

ADENZ: Average Distance between Events and Non-Events via *z*-score transform

PLS: Projection to Lateral Subspaces

GADEN: Genetic Average Distance between Events and Non-events

GADENZ: GADEN via *z*-score transform

RADEN: Random Average Distance between Events and Non-events

RADENZ: RADEN via *z*-score transform

MISFETS: Mixed-Subject Index Feature Selection

MSADEN: MISFETS combined with ADEN

Pattern Recognition Modules:

LDA: Linear Discriminant Analysis

RBF: Radial Basis Function

SVMG: Support Vector Machines with Gaussian kernel

SVMP: Support Vector Machines with polynomial kernel

Classifier Structures:

Xval: Single classifier leave-one-out cross-validation

Bagging: Bagging ensemble with unweighted majority voting

Plurality: A term for specific bagging implementation

Stacking: Stacking ensemble

Boosting: Boosting ensemble

AdaBoost (3 weak learners): Adaptive boosting ensemble with 3 weak learners

AdaBoost (30 weak learners): Adaptive boosting ensemble with 30 weak learners

Primary Dataset Codes:

SABIL: Study A bipolar EEG with ICA preprocessing and log spectral features

SABIS: Study A bipolar EEG with ICA preprocessing and spectral features

SARUS: Study A referential EEG unprocessed by ICA and spectral features

SABUS: Study A bipolar EEG unprocessed by ICA and spectral features

SCRIS: Study C referential EEG with ICA preprocessing and spectral features

“If the brain were so simple we could understand it, we would be so simple we couldn't.”
-Lyall Watson

CHAPTER 1. INTRODUCTION

1.1 Overview

Lapses are breaks in attention and responsiveness for brief durations. Lapses range from brief pauses to microsleeps lasting up to 15 s. Lapses can lead to multiple fatalities in certain occupational fields (e.g., transportation and military) due to the necessities of constant vigilance. A fatigue monitoring and lapse prevention system, able to monitor an individual's state of responsiveness in real time, could assist in the reduction of poor performance and occupational fatalities (Torsvall and Akerstedt, 1987; Jung et al., 1997; Peiris et al., 2006b). Lapses are often difficult to detect, even when multiple types of signals are used (Poudel et al., 2008). Automation of lapse detection is a significant step towards the construction of a lapse-prevention system. The goal of this research was to utilize new methods and refine existing feature extraction and classification techniques to improve EEG-based lapse detection.

A lapse detector shares similarities with another well-documented biosignal feedback device: the brain-computer interface. A brain-computer interface (BCI) provides a direct pathway between a neurophysiological signals and an external device (Blankertz et al., 2008). Electroencephalography (EEG)-based BCI systems require rapid and accurate classification in short periods to provide feedback to the user, thus providing a functional closed loop system. An EEG-based microsleep detector would operate according to the same schematic, and techniques from BCI research might improve the performance of a microsleep detection system.

A particular category of machine learning techniques, proven in BCI, held special relevance to microsleep detection. Supervised machine learning techniques use *a priori* knowledge of class labels to better discern between events in the EEG. Supervised learning techniques in feature selection/reduction (FS/R) could improve the previous system, which relied upon unsupervised FS/R measures, such as principal component analysis (PCA). As BCI systems rely on supervised FS/R and classification algorithms (Blankertz et al., 2008), the potential for performance improvement required investigation.

Beyond investigating new machine learning techniques, the effect on detection of varying the EEG preprocessing and feature extraction steps has not been covered in prior microsleep detection literature. The prior performance benchmark (Peiris et al., 2011), with a mean phi correlation value of 0.39, was achieved using bipolar EEG "cleaned" via artefact

pruning and independent component analysis (ICA). A microsleep detection system operating in real time cannot use ICA to remove artefacts to the same degree as ICA used in offline processing of hour-long sessions. Given the limitation, the effects of not including ICA and artefact pruning were fully investigated in this work.

Specific machine learning techniques exist to deal with imbalanced datasets. Microsleeps are statistically outnumbered by non-microsleep states, forming a highly unbalanced dataset. By training on unbalanced data, a classification system can be biased towards non-events, which comprise the majority of time in microsleep studies (Jung et al., 1997; Peiris et al., 2006b). However, the training data can be artificially balanced by repeating instances of microsleeps and deleting non-microsleep segments, which can potentially remove classifier bias.

An additional consideration is the possibility of EEG-based microsleep prediction. Davidson et al. (2007) and Poudel et al. (2008) raised the possibility of there being spectral changes in the EEG which presage the occurrence of microsleep events. As such, scenarios could be evaluated involving microsleep prediction by EEG spectral features alone. Such research has not been attempted before this work, despite the applicability to microsleep detection.

1.2 Motivation

This thesis demonstrates progress towards the development of a product potentially able to save lives. In addition to its commercial potential, the research is relevant to neural engineering and sleep research. This research represents the continuation of prior work on EEG-based lapse detection (Davidson et al., 2007; Peiris et al., 2011), as well as the revision of previous methods. In addition, investigation of physiological and behavioural lapse detection could prove useful towards the development of a prototype device.

Before a prototype device could be developed, large gaps in the literature for microsleep detection required investigation. The application of BCI algorithms to microsleep detection, the investigation of alternative training scenarios, the variation of preprocessing techniques, and prediction of microsleeps are the primary topics covered.

1.3 Local Research and Expertise

The Christchurch Neurotechnology Research Programme (NeuroTech™), based in the New Zealand Brain Research Institute (NZBRI), has considerable experience in the area of lapse detection. NeuroTech has investigated lapses using EEG, electrooculography (EOG), functional Magnetic Resonance Imaging (fMRI), and behavioural metrics to investigate

subject responsiveness. NeuroTech is closely linked with the Department of Medical Physics and Engineering at Christchurch Hospital, the Departments of Electrical and Computer Engineering, Psychology, and Communication Disorders at the University of Canterbury, and the Department of Medicine at the University of Otago, Christchurch. Expert knowledge regarding the interpretation and processing of EEG, video, and performance data is available. Four expert-rated datasets, comprising EEG and other information, have been collected by NeuroTech and are available to this project.

1.4 Goal

The overall research goal was:

Design a system for improved, automated EEG-based microsleep detection. This included:

a. Increasing the accuracy, sensitivity, and specificity of the detector.
b. Reducing system latency for automated microsleep detection.
c. Determination of optimal spatial and spectral information for accurate microsleep detection.
d. Prediction of the onset of microsleeps project.

1.5 Conceptual Development

- Concepts in this thesis developed from prior work covered in Chapter 2.
- The system implementation is covered in Chapter 3.
- The details of each implemented module are covered in Chapter 4.
- The information on the datasets that were examined is covered in Chapter 5.
- The validation of the system on artificial data is covered in Chapter 6.
- The replication of prior work with the system is covered in Chapter 7.
- Chapter 8 covers the comparative performance of alternative feature sets.
- Chapter 9 covers the elimination of less efficient system configurations.
- In Chapter 10, the effects of artificially balancing data are explored.
- Due to variations in performance being attributable to subject variance, Chapter 11 covers the topic in greater detail.
- Based upon a method of “mixing” features together, Chapter 12 explores this as the basis of a form of feature selection.
- Afterwards, the possible prediction of microsleeps is investigated in Chapter 13.
- Following this, the primary points are mentioned in Chapter 14.

CHAPTER 2. BACKGROUND

2.1 Introduction

In order to understand the goal of the research, relevant microsleep and lapse terminology must be clarified. Due to the breadth of terms in the literature, the definitions of specific, repeated terms in the context of microsleep research were expanded on.

2.2 Lapses of Responsiveness

A lapse of responsiveness is a categorical term for a transient failure to respond while performing a goal-oriented task (Harrison and Horne, 1996; Peiris et al., 2006b). Lapses are the result of several factors within the body and nervous system. The underlying processes behind lapses must be understood in order to understand the causes for lapses. Lapses can be categorized as those due to loss of attention (ability to focus on a task), loss of arousal (based on the physiological state of the body), sleep-wake mechanisms (desire to sleep), and combinations of the previously listed factors. The effects of lapses include response errors (based on errors in planning and execution) (Reason, 1984), delayed responses (when a timed response is necessary) (Williams, 1963), and detection failures (where a changing situation is not accounted for) (Mackworth, 1957).

2.3 Attention

Attention has been hypothesized to comprise multiple components. Descriptions of the specifics of each component vary in the literature. Three of the primary components identified by Posner and Petersen, (1990) are selection, alertness, and capacity. Selection includes attentional orientation, focus, and prioritization of information sources during a task. Alertness possesses two components: general wakefulness (also known as tonic alertness) and ability to temporarily increase response readiness (Parasuraman et al., 1998). Capacity refers to executive attention, the ability to process information when faced with distraction.

Two related components, as identified by Sarter in 2001 (Sarter et al., 2001), are sustained attention and vigilance. Sustained attention is sometimes defined as the ability to respond to frequently occurring events over time. On the other hand, vigilance is sometimes defined as the ability to respond to rare and infrequent events. Sustained attention is of particular interest when studying lapses, due to the probable influences of monotony and sleep deprivation on responsiveness (Sarter et al., 2001).

2.4 Arousal

Arousal refers to a state of cortical activity. The physiological function of arousal is tonic neuronal activity, which alters body physiology. Despite initially being considered a unitary process, some research has indicated multiple pathways affecting arousal (Steriade, 1996). Previous research has found correlations between shifts in lower-level systems and the ability of higher-level systems attempting to compensate for them. The state of arousal is important to understand the physiological underpinnings of lapses (Robbins et al., 1998).

Related to arousal is the dimension of wakefulness. Wakefulness refers to a state of alertness that promotes attentiveness. Alert phases are characterized by excitability and attentional control. On the other end of the spectrum is sleep promotion. Sleep is a complex physiological process, comprising multiple stages. A commonly used standard is the Rechtschaffen and Kales (R&K) scale, which includes W (wakefulness), non-REM (NREM or non-rapid eye movement) sleep (with stages 1, 2, 3, and 4), and REM sleep (Moser, 2009). EEG changes measure the transition between sleep stages (Jung et al., 2010). As a subject moves from awareness to sleep, drowsiness sets in and information processing capacity declines.

2.5 Lapse Categories

Transient lapses in performance can be due to temporary disruptions in the brain (Harrison and Horne, 1996; Peiris et al., 2006b). Lapses can be broadly separated into attention lapses and arousal lapses. Arousal-based lapses can be sorted into different categories based on certain criteria. Arousal-based lapses also possess the physiological and behavioural signs of drowsiness and sleep (Chee et al., 2008). A behavioural microsleep ("microsleep") is a type of lapse where the lack of response to a task lasts from 0.5 to 15 s, full or partial (>80%) eye closure, and drowsy behaviour (Lal and Craig, 2001; Peiris et al., 2006b; Golz and Sommer, 2010). Microsleeps can occur even in well-rested individuals (Peiris et al., 2006b; Innes et al., 2010; Jones et al., 2010; Poudel et al., 2014). Another type of lapse is the lapse of task-oriented attention (Jones et al., 2010). It includes a complete diversion of attention of >0.5 s. A voluntary eye closure (VEC) may be performed during such a lapse when a subject is fatigued (perhaps for temporary relief). In lapses of sustained attention, the lack of response is greater than 0.5 s with no eye-closure other than normal blinks, and is unrelated to the level of arousal but due rather to changes in attention (Jones et al., 2010).

2.6 Electroencephalography

EEG is the electrophysiological measurement of neural function via scalp electrodes. The first EEG experiments were performed in the late nineteenth century and early twentieth century (Jung et al., 1997). The technology is used primarily in hospitals and in medical research. Medical uses of EEG include detecting signs of mental activity in catatonic patients (Ward et al., 1999), distinguishing epileptic seizures (Hoeve et al., 2001), and many related applications (Othman et al., 2009). EEG is commonly used in psychology, neuroscience, and cognitive science research. The most commonly used pattern of EEG electrode placement is the International 10-20 System. Electrochemical activity from neuronal discharge results in the signal measured by non-invasive electrodes (Duffy, 1989). The amplitude of the signal is low, typically measured in microvolts, and is commonly amplified by a factor of a thousand into millivolts (mV) before processing. As such, EEG is sensitive to ocular and muscular artefacts. EOG, electrophysiological measurement of eye movement, is often taken with EEG to help remove ocular artefacts. Features of interest in EEG are typically low-frequency (<100 Hz), so the Nyquist sampling criteria can be easily fulfilled to prevent aliasing.

EEG can be broken into several frequency bands. Frequency bands include the delta (0-4 Hz), theta (4-7 Hz), alpha (8-13 Hz), beta (>13-30 Hz), and gamma (30-100 Hz). Related to the alpha band is the mu band (8-13 Hz), present in the mirror neurons of the sensorimotor cortex and is often studied in movement science. EEG has been used for decades in sleep research (Jung et al., 1997). Changes in sleep have been defined by changes in different bands (Hori et al., 1994). Microsleeps have also been correlated with changes in theta band activity (Poudel et al., 2010). EEG is non-invasive and cheaper than other types of bioinformatics technologies; researchers have studied EEG-based lapse detection for several years, but with limited success (Peiris et al., 2006a; Davidson et al., 2007; Peiris et al., 2011).

2.7 Lapse and Microsleep Detection

Both physiological and behavioural measurements are used in lapse research. Physiological measurements are those directly dependent upon neurophysiological events, including EEG, fMRI, and EOG (Golz et al., 2005). Behavioural measurements do not depend directly on a specific physiological parameter, and include performance on behavioural tests and video recordings of eye movements (Krajewski et al., 2008). Physiological recordings offer more information about neural and physiological function, but are more suitable for a laboratory or clinical environment. Behavioural measurements offer less information about internal activity, but are more suitable for use outside of a research

environment and frequently form the "gold standard," or rating system, for such an event (Peiris et al., 2004a; Bergasa et al., 2006; Peiris et al., 2006b). Various combinations of physiological and behavioural metrics have been used in several earlier studies (Davidson et al., 2007; Krajewski et al., 2008; Poudel et al., 2010).

The related fields of fatigue/drowsiness estimation and sleep detection provided relevant data for lapse detection (Valley and Broughton, 1983; Torsvall and Akerstedt, 1987; Conradt et al., 1999b; Van Orden et al., 1999; De Gennaro et al., 2000; Doran et al., 2001; Vuckovic et al., 2002; Zocchi et al., 2007).

A potentially related area to EEG-based microsleep prediction is detection and prediction of epileptic activity. Epileptic seizures are defined by a "spike" in EEG, as well as high frequency activity (Hoeve et al., 2001). Techniques used in microsleep research have also been applied to epileptic activity detection, such as spatial filtering and wavelets (Goelz et al., 1999). Spectral features were also used in seizure prediction (Park et al., 2011). However, microsleeps lack the EEG spikes of epileptic activity and high frequency activity. As such, techniques successfully applied to epileptic activity detection may not have the same success with microsleep detection, and vice versa.

Much of the relevant research in microsleep detection has involved correlating sleep and drowsiness to electrophysiological signals, such as EEG or EOG. Due to the variety of studies and approaches, inconsistencies may arise. Lapse detection by human experts based on behavioural data is time-consuming, necessitating the automation of the process (Peiris et al., 2005b). Sometimes, even human experts are uncertain whether behavioural measures indicate a microsleep. Some ambiguous portions of the EEG datasets have been included in prior training sessions, likely confusing the classifier. The ambiguous segments will be reclassified or removed and the performance of the detection system tested to see whether this leads to increased accuracy.

2.7.1 Electrophysiological Measurements

2.7.1.1 EEG

EEG has been utilized for decades in sleep research, although the methods and standards differ widely (Peiris et al., 2008). EEG is also non-invasive and commonly used in medical and research environments. Issues with EEG include signal acquisition and low spatial resolution compared to other methods, such as fMRI (James et al., 1996). EEG is highly sensitive to ocular and muscular artefacts. Additional processing, often utilizing

independent component analysis (ICA), is often necessary to remove ocular artefacts (Davidson et al., 2005; Peiris et al., 2005a; Peiris et al., 2006a; Peiris et al., 2011).

The sleep state is traditionally associated with more complex and variable EEG states than the awake state (Davidson et al., 2007; Jung et al., 2010; Poudel et al., 2010). However, it is difficult to determine the difference between both states over short periods. Even a given individual will demonstrate a variety of different EEG patterns during the transition between states. Lapses may occur over short periods, providing little time for a system to identify a lapse (Peiris et al., 2011).

For these reasons, EEG-based automatic lapse detection is a complex signal processing problem. A lapse detection system includes three essential steps: preprocessing, feature extraction, and pattern recognition. Preprocessing removes artefacts from the raw data. Feature extraction algorithms separate desired information from background noise and select optimal features, such as power of EEG spectral bands. Pattern recognition techniques assign feature instances into categories based on prior training.

Previously investigated preprocessing and feature extraction techniques applied to EEG-based detection of lapses and microsleeps include power spectral density (PSD) (Peiris et al., 2006a), power ratios (Peiris et al., 2011), fractal dimensions (FD) (Peiris et al., 2005a), similarity indices (Poudel et al., 2010), Lempel-Ziv complexity index (LZ) (Peiris et al., 2011), wavelets (Goelz et al., 1999), delayed vector variance (DVV) (Golz et al., 2007), modified periodograms (Golz et al., 2007), bispectral indices (BIS) (Pomfrett and Pearson, 1996; Greenwald et al., 1999), ICA (Peiris et al., 2011), principal component analysis (PCA) (Peiris et al., 2011), spectral coherence analysis (SCA) (Dehbaoui et al., 2011), approximate entropy (ApEn) (Peiris et al., 2011), and spectral entropy (Peiris et al., 2011).

It is not uncommon to use several sets of features in lapse detection and related areas (Peiris et al., 2011). Feature extraction techniques can be used in combination with each other. The purpose of multiple feature extraction steps is to utilize one or more as a method of preprocessing (if performed sequentially) or to extract different parameters from the same data (if performed in parallel). As a result, matrices of features can have high dimensionality and complexity. Certain techniques are used to reduce the dimensionality of a feature set, such as PCA (Peiris et al., 2005a; Peiris et al., 2006a; Peiris et al., 2008; Peiris et al., 2011). Spatial mapping algorithms may be potentially useful in pattern recognition (Hoeve et al., 2001).

Previously investigated pattern recognition techniques for lapse detection and related areas include neural networks with back propagation (James et al., 1996; Vuckovic et al.,

2002), fuzzy-logic-based classifiers (Coufal, 2009), self-organizing maps (Golz et al., 2001; James et al., 1999), support vector machines (SVM) (Golz and Sommer, 2010), long short-term memory (LSTM) recurrent neural networks, and linear discriminant analysis (LDA) (Davidson et al., 2007). The accuracy of an automated detector varies between the types of classifiers, although performance increased in an LSTM compared to simple network architectures and classifiers (Kirk and LaCourse, 1996; Davidson et al., 2005; Krajewski et al., 2008).

Previous studies have investigated drowsiness and fatigue in relation to the use of EEG (Kiymik et al., 2004). Effective EEG-based detection of microsleeps has proved a complex problem. Two previous studies have indicated an EEG-based alertness estimation system using as few as two electrodes was feasible (Jung et al., 1997). A neural network trained on spectral coefficients of a Discrete Wavelet Transform (DWT) was able to indicate if a subject was drowsy (Kiymik et al., 2004). Support Vector Machines (SVM) have been used to classify fatigue-related features from a combination of EEG and EOG (Golz and Sommer, 2010).

In previous lapse studies by NeuroTech, ICA and filtering were used on EEG data to remove ocular artefacts and noise (Davidson et al., 2007; Peiris et al., 2005a; Peiris et al., 2006a; Peiris et al., 2011). Various other features, including FD (Peiris et al., 2005a), LZ (Peiris et al., 2011), and spectral coefficients (Peiris et al., 2006a), have been used to detect lapses and microsleeps. However, accuracy of previous automated detectors did not meet desired levels. Much of NeuroTech's research has involved combinations of EEG and other data. EOG was taken along with EEG to assist with artefact pruning.

2.7.1.2 EOG

Eye movements and closures can be measured with EOG. With EOG, it may be possible to detect drowsiness through measurements of slow eye movements (SEMs)(De Gennaro et al., 2000; Leong et al., 2007). However, EOG requires electrodes to be placed around the eyes, which can cause subject discomfort over extended periods (Alba et al., 2010). Similar to EEG, EOG is primarily used in a laboratory or clinical setting. In prior lapse and microsleep detection research, EOG was utilized to remove ocular artefacts (Peiris et al., 2008; Peiris et al., 2011).

2.7.2 Behavioural Measurements

Lapse detection has included non-electrophysiological measurements, such as behavioural data and video recording of eyes (Peiris et al., 2004b; Bergasa et al., 2006; Golz and Sommer, 2010). Video-based systems can be used to detect the timing and duration of eye closure events (Bergasa et al., 2006). Behavioural test performance offers another avenue for lapse detection (Doran et al., 2001). Different types of behavioural measurements are often coupled together to achieve a detection result (Peiris et al., 2004b).

A common type of behavioural measurement is performance on a particular task (Makeig and Inlow, 1993; Poudel et al., 2010), such as one requiring continuous attention (Valley and Broughton, 1983; Van Orden et al., 1999; Peiris et al., 2006b). A previous microsleep detection system used speech, but was impractical outside of a research environment due to reliance on speech samples (Krajewski et al., 2008). Task performance can be combined with physiological signals to detect lapses (Peiris et al., 2008). A system that does not require electrophysiological signals would be more practical for an occupational environment due to cost and ergonomics, although accuracy may suffer.

2.7.2.1 Task Performance

Behavioural tests are common in drowsiness research (Dinges and Grace, 1998; Bergasa et al., 2006). A common type of test is the psychomotor vigilance task (PVT) (Dinges and Powell, 1985; Anderson et al., 2010). During a PVT, a subject responds to cues 2-10 s apart over 10 min. The drive to sleep, including the ability to respond to signals and pay attention, can be captured by the PVT (Dorrian et al., 2005). A related behavioural test type is the Reaction Time Test (RTT), where the response latency of a subject is measured (Conradt et al., 1999a). Another type of task is one requiring continual attention and performance from the subject, such as a continuous tracking task (CTT), where a subject must move a marker towards a shifting target (Peiris et al., 2006b). Tracking tasks used in prior lapse detection studies have been one dimensional (Peiris et al., 2005b; Davidson et al., 2007; Peiris et al., 2011) and two dimensional (Poudel et al., 2008; Innes et al., 2010; Poudel et al., 2010). Performance tests have been used to detect lapses, based upon performance stopping during microsleeps (Poudel et al., 2010). Behavioural and cognitive tasks require a user to be constantly performing them, limiting their applicability in the field due to unpredictable breaks in routine.

2.7.2.2 Video

Video recordings of a subject's eyes are used to measure SEMs and eye closure events associated with drowsiness (Malla et al., 2010). One feature used to detect eyelid movements on video is the mean percentage of eye closure over 1 min (PERCLOS) (Bergasa et al., 2006; Malla et al., 2010). Computer vision estimates the size of an individual's pupil. As the eyelid closes, less of the pupil is visible to the camera, altering the value of the PERCLOS feature (Dinges and Grace, 1998). The quality of recordings across studies varies greatly based on a number of parameters, such as video quality, visible spectrum, frame rate, distance from the lens to the subject, angle of focus, and ambient lighting. High quality recording allows a greater chance for successful extraction of the PERCLOS feature.

Previous studies indicate that PERCLOS-based computer vision may be sufficient for detecting drowsiness under ideal conditions (Bergasa et al., 2006; Hanowski et al., 2007; Malla et al., 2010). However, PERCLOS is typically measured over a minute. The time required for lapse detection is much shorter. One study integrated performance on a tracking task with PERCLOS-based computer vision (Malla et al., 2010). The result was a highly sensitive program that measured flat points in a tracking task and video data, but had a high rate of false positives (Malla et al., 2010). Combinations of metrics offer the advantage of combining several types of data at once. Despite limited success, the technology requires improvement before suitable for usage in a commercial product (Bergasa et al., 2006).

The use of PERCLOS and other video-based methods for lapse detection has a number of technical issues in a real-world environment, as problems exist with both hardware and software. Hardware issues include the positioning of the camera, frame rate, data processing speed, design of the detector, ergonomics, noise, and lighting affecting the quality of the features. Software issues include detection of the pupil, ambient lighting, eyelashes interfering with detection, and demands for high-speed data processing. Even in the prototype phase, these problems have not yet been fully addressed. For these reasons, an EEG-based microsleep detector offers an alternative with a strong basis in research (Peiris et al., 2011).

2.7.3 Combination of Physiological and Behavioural Data

Several studies combined physiological and behavioural data to identify the transition between alert and sleep states (Peiris et al., 2005b; Krajewski et al., 2008; Peiris et al., 2011). EEG features are often used with other measures in lapse detection studies: studies from NeuroTech combined EEG with eye-video and performance on a tracking task (Davidson et al., 2007; Jones et al., 2010; Peiris et al., 2011). One lapse study examined lapses in 15

normal subjects performing a CTT (Peiris et al., 2004a). EEG, EOG, video recordings, and CTT performance were measured. Following this, EEG-derived PSD coefficients were used with long short-term memory (LSTM) neural networks. The neural networks were trained to detect lapses, but did not perform reliably (Davidson et al., 2007). All studies from the NeuroTech group have human experts identify and rate lapses in the data sets, which set a benchmark for any lapse detection algorithm (Peiris et al., 2005b).

The following studies examined different methods of feature extraction. EEG data was primarily utilized, with the EOG and video serving to assist with verifying lapses. One paper examined the possibility of using FD as its chief feature extraction method (Peiris et al., 2005a). However, this produced few useful results. Thus, FD was found to be ineffective for microsleep detection (Peiris et al., 2005a). Spectral power of EEG frequency bands and power ratio features were also examined. However, performance was only modest for microsleep detection (Peiris et al., 2006a). Neural networks with varying architectures, such as a tapped delay line (TDL) linear perceptron and LSTM system, classified data based on a sliding feature window (Davidson et al., 2007). Performance with spectral-based features was satisfactory, and the LSTM network performed better than the TDL system in lapse identification. A recent study examined PSD and compared it with ApEn, FD, and LZ complexity (Peiris et al., 2011). Spectral features performed better than the others (Peiris et al., 2011). This paper provided the current benchmark for detection of microsleeps, with an accuracy (61.2%), a reasonably high sensitivity (73.5%), but a low selectivity (25.5%).

Another study combined EEG and fMRI data to determine the relationship underlying microsleeps in the brain. A 2D CTT was developed for the study (Poudel et al., 2008). The data from the study included fMRI, EEG, EOG, eye videos, and tracking task performance (Jones et al., 2010). Further analysis yielded findings demonstrating a correlation between theta band power and visuomotor error (Poudel et al., 2010).

Related research in the fields of fatigue and drowsiness have used other types of feature extraction and pattern recognition. Support vector machines (SVM), a kernel based algorithm, has rarely been applied to EEG-based automatic lapse detection (Golz and Sommer, 2010). Previous studies (Golz and Sommer, 2010) used a combination of biometric signals (EEG and EOG) alongside self-reported sleepiness. The study had reasonable success, with a mean error rate of 9% over 22 subjects.

2.7.4 Limitations of Previous Approaches

Despite several years of research, no system has been able to accurately and consistently detect microsleeps using EEG data or other physiological metrics. Some approaches in lapse detection systems have gone towards behavioural measurements, such as video and task performance (Bergasa et al., 2006; Zocchi et al., 2007; Krajewski et al., 2008). In addition, a video-based lapse detector has been designed (Malla et al., 2010). This is highly sensitive, but additional filtering may reduce the number of false positives and increase specificity and accuracy.

An innate limitation in EEG-based microsleep detection is the relevant brain-states responsible are highly variable and speculative. As a result, ratings based on video and behavioural recordings were used to compensate for this, but both present possible shortcomings as estimates of brain-state (Peiris et al., 2011). The quality of the EEG electrode connections and impedances can vary between sessions and even over the course of the same session, limiting efforts to quantify the relevant brain-states (Othman et al., 2009).

While EEG signal quality may be "improved" through use of filtering, baseline removal, artefact rejection, and similar methods, doing so could distort relevant information. The exact brain-state of a given microsleep is still very speculative, given that the EEG recorded during a microsleep can be highly variable in quality, and that any features derived from that EEG segment may also be similarly variable in quality. Likewise, the use of a human expert can introduce other uncertainties. As such, EEG-based microsleep detection tends to be challenged by very noisy and imbalanced data (Davidson et al., 2007; Peiris et al., 2011).

A disadvantage with EEG-based systems is the requirement that an individual must continuously wear EEG electrodes, which may be impractical for a person in certain environments. A video-based system, by contrast, does not depend on electrodes. While video has its advantages and disadvantages, an EEG-based system is still viable (Peiris et al., 2011). For EEG-based detection, the system must be able to perform accurately and in real time. Several feature extraction and classification algorithms which have been found to be useful in other applications have not yet been applied to the EEG-based lapse detection problem. The requirements of EEG-based detection mirror demands in other fields, which may assist in performance improvements.

2.8 Relevant Techniques from BCI

EEG-based microsleep detection requires accurate and rapid classification of neural signals. The system must be able to perform accurately and in real time, its requirements mirroring demands in other fields. Several algorithms from other fields have not yet been applied to EEG-based microsleep detection, and may improve performance. BCI algorithms require rapid and accurate classification of neural signals. BCI is the direct use of neurophysiological signals, including EEG, to control an external device (Blankertz et al., 2008). Classification in BCI must occur within a period of approximately 200 ms, the latency period of human awareness (Blankertz et al., 2008). In BCI, faint EEG-based features must often be separated from noise rapidly and accurately. Features could vary highly across individuals.

2.8.1.1 Feature Extraction Algorithms

As such, certain algorithms for feature extraction and pattern recognition in BCI may be applicable to EEG-based lapse detection. A microsleep detector essentially functions as a BCI based upon involuntary events. Not all BCI algorithms are relevant to lapse detection. Certain features used in BCI (Quitadamo et al., 2009; Lijing et al., 2012), such as evoked potentials and event-related potentials (ERPs), are unsuitable for use in lapse detection. Other features have been successfully utilized in both BCI and lapse detection, such as spectral features (Blankertz et al., 2008; Dobrea et al., 2010; Peiris et al., 2011).

EEG spectral features may incorporate both spatial and temporal data. These spatio-temporal features can include wavelets and information on correlations between channels, and have worked with fuzzy logic and neural classification systems (Kasabov and Song, 2002). Spatio-temporal features have already been successfully used in seizure detection (Chavez et al., 2003), time-series prediction (Kasabov and Song, 2002), and BCI (Lakany et al., 2006). However, wavelet transformation features did not improve performance on epileptic spike detection (Goelz et al., 1999). For these reasons, the use of spectral features concatenated in a vector was considered adequate.

For the benchmarks in microsleep detection, the primary feature extraction method used in the prior benchmarks was a matrix of 544 spectral features calculated for every 2 s, with 34 power spectral features derived 16 bipolar channels (Davidson et al., 2007; Peiris et al., 2011). Spatio-temporal information was potentially lost by concatenating information together into a single vector in the feature matrix, but the use of spatio-temporal features like wavelets did not improve performance (Goelz et al., 1999). In contrast, the use of spatial and

spectral features offered the potential to determine changes in brain-state by isolating specific electrode channels and spectral bands of interest. Additionally, the matrix of spectral features was sufficient to achieve the prior benchmarks in microsleep detection (Davidson et al., 2007; Peiris et al., 2011).

2.8.1.2 Feature Selection and Reduction Algorithms

For feature selection and dimensionality reduction, PCA has been successfully utilized in both BCI and lapse detection (Selim et al., 2009; Peiris et al., 2011). Supervised feature selection and reduction methods offered a promising direction to investigate (Omary, 2009; Raudys, 1991). Common spatial patterns (CSP) is a method of supervised learning and has offered increases in performance over PCA (Lu et al., 2009), but has not has been applied to the lapse detection problem. Projection to latent subspaces (PLS) is a method of supervised learning and has been successfully used to find evoked potentials in EEG (Chen, 2013; Hutapea, 2014), but not applied to lapse detection. Genetic algorithms (GAs) have been successfully used in BCI (Parini et al., 2007; Wang et al., 2011), but not lapse detection. GAs could find optimal combinations of features to increase classifier performance, as was performed in BCI research.

Another aspect of FS/R relevant to microsleep detection is the supervised selection of the most informative features. To select informative features, a trade-off exists between bias and variance, after which additional features become redundant. Mutual information theory can reduce redundant variables to informative ones (Reshef et al., 2011). Introduced by Reshef et al. (2011), maximal information coefficient (MIC) was one of several maximal information-based nonparametric exploration (MINE) methods intended for use in finding the most relevant parameters. Related techniques aim to eliminate redundant information to generate an informative subset of features, such as minimal redundancy, maximal-relevancy (mRMR), as proposed by Peng et al. (2005). Many of the algorithms operate by binning, or sorting variables into larger "bins," including specific categories like EEG time windows (Zheng et al., 2010).

Arguably, the previous lapse detection benchmark organized spectral information into bins based upon EEG spectral bands. Concentrating spatial and spectral features for a 2-s window into a single vector potentially lost relevant information. As such, it was decided to explore distance correlation of spectral features, as to find features with the greatest differences (Székely et al., 2007) and correlation (Székely and Rizzo, 2009). It was decided to continually adjust the number and type of features retained, similarly to GA and greedy

search algorithms (Hazewinkel, 2011). Such iterative algorithms improved BCI and machine learning performance (Kim et al., 2006; Parini et al., 2007; Wang et al., 2011).

2.9 Relevant Techniques from Machine Learning

Related directly to BCI is the field of machine learning. For pattern classification techniques, LDA has been utilized in both lapse detection and BCI (Gareis et al., 2011; Peiris et al., 2011). Other pattern recognition algorithms have been utilized in lapse detection and BCI, including self-organized maps (SOMs) (Golz et al., 2001; Sommer et al., 2001; Yamagutchi et al., 2007) and support vector machines (SVMs) (Ruping, 2001; Golz et al., 2007; Krajewski et al., 2008; Golz and Sommer, 2010). A radial basis function (RBF) has been successfully utilized in BCI and other areas for classification, and can perform better than a traditional neural network in some applications (Finan et al., 1996). RBFs have not yet been utilized in lapse detection, although a related fuzzy-logic based classifier has successfully been used with biosignals (Geva, 1998). The algorithms that were considered for further investigation are PLS, CSP, GAs, SVMs, and RBFs.

2.9.1.1 Classifier Ensembles

In addition to single classifiers, ensembles were applied to microsleep and lapse detection. By combining multiple classifiers, an ensemble of them can achieve better performance than single classifiers (Opitz and Machin, 1999). Ensembles have been used with neural data before (Honorio et al., 2012) and used in BCI (Shoaie et al., 2006; Faradji et al., 2010). The previous benchmark (Peiris et al., 2011) used a stacked generalization (stacking) ensemble (Wolpert, 1992).

The type of ensemble that achieves the best performance can be circumstance specific (Zenko et al., 2001), and research may determine if one type of ensemble can consistently perform better than others on a particular dataset. The four ensembles were considered are bootstrap aggregating (bagging) (Breiman, 1996), boosting (Schapire et al., 2005), adaptive boosting (AdaBoost) (Freund and Schapire, 1997), and stacking (Wolpert, 1992).

2.9.1.2 Balanced Data

Even the performance of a classifier ensemble could be affected by unbalanced data, where one class comprises a much smaller percentage of the other class. As a result, a classifier could be biased due to seeing more examples of the larger case. As such, a classifier could be trained on artificially balanced data. A dataset could be balanced by randomly deleting instances of the majority class and repeating instances of the minority class (Raudys,

1991). Due to unbalanced microsleep data, classifier bias may affect performance. Therefore, artificially balanced microsleep data was considered for further investigation.

2.10 Specific Goals

The goal of this research project was to explore new methods of automating microsleep detection and, in the process, increase the accuracy of detection. Two datasets previously recorded, each including EEG and other information (such as EOG, video recordings, and performance data) were examined and different types of signal processing, feature extraction, and pattern classification algorithms applied differently than in previous studies. Microsleeps are accompanied by changes in the EEG, and the development of a feature-based detector may predict the occurrence of a lapse early enough to avoid the occurrence.

As restated from Chapter 1, the overall research goal and steps were:

Design a system for improved, automated EEG-based microsleep detection. This included:

a. Increasing the accuracy, sensitivity, and specificity of the detector.
b. Reducing system latency for automated microsleep detection.
c. Determination of optimal spatial and spectral information for accurate microsleep detection.
d. Prediction of the onset of microsleeps.

In order to accomplish these objectives, the following specific steps were taken:

a. Design of a modular software detector toolset (Chapter 3).
b. Investigation of techniques to reduce data required for successful (Chapter 4).
c. Generation of different feature sets for examination (Chapter 5).
d. Evaluation of software toolset on artificial data (Chapter 6).
e. Replication of earlier benchmarks (Chapter 7).
f. Evaluation of the feature set variants warranting further research (Chapter 8).
g. Reduction of the total number of system configurations investigated to the most promising (Chapter 9)
h. Exploration of the effects of altering class balance on training and testing (Chapter 10).
i. Determination of optimal training circumstances by exploring various combinations (Chapter 11).

j. Application of a supervised feature selection method as a preprocessing method (Chapter 12).
k. Examination of the potential for microsleep prediction using techniques shown (Chapter 13).

CHAPTER 3. SYSTEM IMPLEMENTATION

3.1 Overview

A complete lapse detection system involves preprocessing, feature extraction, feature reduction/selection, and classification steps, as shown in Figure 3.1. The preprocessing step includes signal acquisition, filtering, and artefact pruning. The feature extraction step takes the processed EEG data and returns a set of features based upon an algorithmic process. More than one set of features can result from one set of data, forming a matrix of different types of feature sets. The number of features is reduced/selected in various ways, such as PCA, so as to minimize and optimize the number of features given to the classifier without losing key information in the feature set. Fewer features reduces the computational complexity and improves the system response times. The final step is pattern recognition. Based upon prior training, each set of features is assigned a category based on the classification algorithm. In these respects, the system is similar to a BCI. A microsleep detector can be considered a BCI for involuntary events.

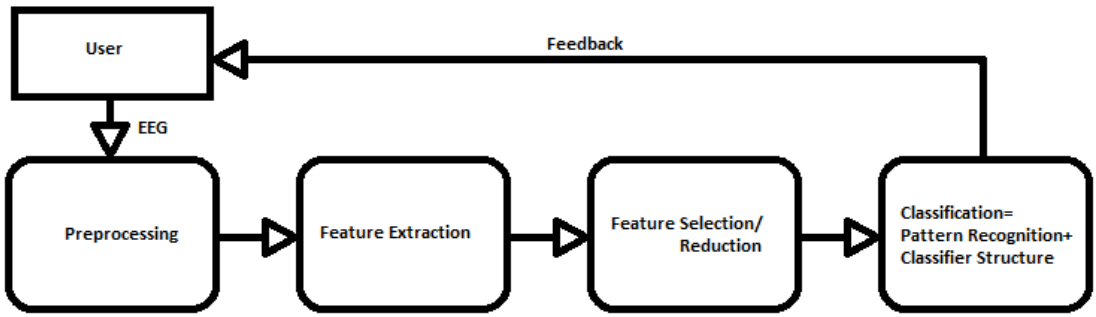

Figure 3.1: Lapse detection system overview

The first step is the implementation of a system to train, test, and validate the performance of an algorithm. The first EEG dataset to be investigated, after the artificial data, was the Study A dataset (N = 8). Leave-one-out class-validation (LOOCV) was used to measure the accuracy of each system, where one subject is used for testing and the rest are used for training the classifier. Each system comprised a different arrangement of feature extraction, feature selection, classification techniques, and training based on different subsets of data. Following this, the procedure was replicated with other EEG datasets. Specific feature extraction and pattern recognition techniques investigated for research are outlined in Chapter 4.

3.2 Implementation

MATLAB was used to analyse the datasets. Algorithms were implemented in MATLAB R2010a on a laptop running Windows 7. Code was implemented in modules, each a part of a modular toolset. The toolset was named the Integrated Canterbury Open Modular Inventory (ICTOMI), and included documentation on each function. Feature extraction, FS/R, pattern recognition, and classifier ensembles all possess different modules. Feature extraction modules include linear spectral features and log spectral features. Feature selection modules include PCA, CSP, PLS, and ADEN-based methods. Pattern recognition modules include LDA, SVMs, and RBFs. Ensemble modules include boosting, bagging, AdaBoost, and stacking. An arrangement of modules represents a specific system configuration. Performance in different configurations was compared with each other. To test and verify the proper functioning of modules, artificial data was utilized. This initially consisted of sinusoids with added Gaussian noise. The frequencies were known in each set of dummy data, to assist in debugging the code. Subsequently, a new category of artificial data was devised to provide a more realistic test. The artificial data combined pre-recorded EEG with artificial events of varying signal-to-noise ratios (SNRs). The use of artificial data is further detailed in Chapter 6. After the code had been verified with artificial data, the EEG datasets were examined. The process of rating microsleeps is further detailed in Chapter 8.

3.3 Testing

Testing was conducted according to prior research (Peiris et al., 2011). Four architectures of classifier ensembles were investigated in addition to single classifier configurations: bagging, boosting, AdaBoost, and stacking. For each classifier system, leave-one-out cross-validation (LOOCV) was used for testing performance. The output was a binary matrix indicating whether each instance in time, at 1-s intervals, was in a microsleep state or not. Then, the performance of the ensemble was evaluated on the validation subject. The procedure was repeated until each subject in the dataset has been utilized as the testing subject. The performance metrics were then averaged together for the final results.

3.4 Performance Metrics

The five main performance metrics that have been used are accuracy, sensitivity, specificity, selectivity, and phi correlation. All five can be calculated from four values: true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*). True positives are correct identifications of events. True negatives are correct identifications of non-events. False positives are incorrect labelling of a non-event as an event. False negatives

are incorrect assessment of an event as a non-event. The methods of calculating all performance metrics are detailed in Equations (1) through (5), and are widely use in machine learning (Omary, 2009).

The accuracy (*Acc*) is the total rate of correct identifications as a percentage of total responses.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

Sensitivity (*Sens*) is the correct response percentage of all positive identifications.

$$Sens = \frac{TP}{TP+FN} \qquad (2)$$

Specificity (*Spec*) is the percentage of correct responses for all negative identifications.

$$Spec = \frac{TN}{TN+FP} \qquad (3)$$

Selectivity, also known as positive predictive value (*PPV*), is the probability of positive identifications being correct.

$$PPV = \frac{TP}{TP+FP} \qquad (4)$$

Phi correlation (*Phi*) is a performance metric which is largely independent of group distributions in a dataset, making it useful for highly imbalanced data, such as the unbalanced artificial data and the Study A dataset (Peiris et al., 2011). The phi correlation is the Pearson correlation coefficient between two binary variables and ranges from 1 to -1. A phi value closer to 1 indicates correct classification, a value closer to -1 means opposite responses than the correct ones, and a value closer to 0 indicates random guessing.

$$Phi = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TN+FN)(TP+FN)(FP+TN)}} \qquad (5)$$

Performance metrics similar to phi exist, such as the kappa coefficient, and convey information in a similar manner (Omary, 2009). In microsleep detection, successful classification of events is valued far more highly than non-events. As a result, sensitivity and selectivity are considered more important than specificity. Due to the small number of events, a high rate of *Acc* can be misleading, as it can be achieved simply by arbitrarily labelling all testing scenarios as non-events. This is why the other metrics were used.

The chance that a classifier might incorrectly label all non-events as events was why the phi correlation was used in conjunction with sensitivity and selectivity. All five metrics were recorded for each configuration of modules, so that direct comparisons of their performance can be made. In cases where two classifiers exhibited similar phi values, the other metrics could provide additional information.

CHAPTER 4. ALGORITHMS INVESTIGATED

4.1 Feature Extraction

Feature extraction converts raw or processed data into a form useable to the microsleep classification system. Prior work (Peiris et al., 2011) utilized features derived from EEG spectral bands. Performances of previous and new feature types were compared. Two feature extraction techniques discussed here are linear spectral features and log power spectral features.

Peiris et al. (2011) used log power spectral features. Linear spectral features were explored but not documented. Both are calculated in similar ways, but both used the same spectral band power and power ratio features. The spectral band power features were calculated first, and then power ratios were calculated.

The following features were generated every 1.0 s for each overlapping 2-s epoch of EEG on each channel. In addition, 12 spectral features normalized with respect to the power of the entire spectrum were included with the 13 spectral features and 9 power ratios for a total of 34 features per channel for each 2-s window, as shown in Table 4.1.

Table 4.1: Spectral features extracted from EEG

Feature	Frequency Band
a) Band Power	
Delta (δ)	1-4 Hz
Theta (θ)	5-8 Hz
Alpha 1 (α_1)	8-10 Hz
Alpha 2 (α_2)	10-12 Hz
Alpha (α)	8-12 Hz
Beta 1 (β_1)	13-16 Hz
Beta 2 (β_2)	16-26 Hz
Beta (β)	13-26 Hz
Gamma 1 (γ_1)	26-36 Hz
Gamma 2 (γ_2)	36-46 Hz
Gamma (γ)	26-46 Hz
High	45-100 Hz
All frequencies	1-100 Hz
b) Power Ratios	
θ/β , θ/α, α/β, δ/θ, α/δ, β/δ, β_1/α, β_2/α, β_1/β_2	Varies

4.1.1 Linear Spectral Features

The power spectral density (PSD) of a signal can be estimated in a number of ways. Taking the power of a signal is common in EEG analysis, and can be accomplished to estimate the composition and power of different frequency bands. In previous work (Peiris et al., 2011), autoregressive spectral estimation was used with the Berg algorithm to calculate Least Mean Squares (LMS) to estimate the power of various spectral bands for each 2-s segment. Each 2-s segment had 50% overlap with the prior segment, and updated at a rate of 1 s. The 2-s segment was thought to be sufficient for classification purposes (Peiris et al., 2011).

Linear Spectral Features Calculation

1) Take 2 s of EEG data from a single channel.
2) Estimate PSD from 1-100 Hz.
3) Calculate all 34 features and put results in a vector.
4) Repeat for all other channels.
5) Concatenate all features as a single observation for 2-s window.
6) Move window forward by 1 s.
7) Repeat for entire duration of recorded EEG.

Alternative methods include the periodogram, based on a finite Fourier transform, and the Welch method of averaging several overlapping periodograms. In this research, spectral estimation was used to calculate PSD features (e.g., alpha band power) from different spectral bands across 16 channels in the case of Study A. A total of 34 features per channel were generated, yielding a total of 544 features per 2 s segment for Study A. For Study C, empty channels were either replaced with interpolated ones or filled with a vector of 34 zeroes, depending on the variant feature set.

4.1.2 Log Power Spectral Features

The log power spectral features were an expansion on the original method of spectral estimation. The highest mean phi performance metrics from prior work (Peiris et al., 2011) corresponded directly to features derived from the natural logarithmic transform of the power spectrum, after it is estimated. The estimate of each spectral band for each 2-s segment was

used to calculate a total of 34 features for each channel. As with spectral features, the sliding window updated at a rate of 1 s.

Log Power Spectral Features Calculation

1) Take 2 s of EEG data from a single channel.
2) Estimate PSD from 1-100 Hz.
3) Perform a natural logarithmic transform of power spectrum.
4) Calculate all 34 features and put them in a vector.
5) Repeat for all other channels.
6) Concatenate all features as a single observation for 2-s window.
7) Move window forward by 1 s.

In Study A, a total of 16 channels yielded 544 features total per 2-s segment, as with the prior spectral feature estimation method. As done previously for Study C, empty channels were either replaced with interpolated ones or filled with a vector of 34 zeroes, depending on the variant.

4.2 Feature Selection/Reduction

The large volume of data and features generated from EEG contains noise alongside relevant information. In previous lapse detection research, a large feature matrix was generated (Peiris et al., 2011). Larger numbers of features are reduced to a smaller number, reducing complex data to more informative representations. Feature selection and feature reduction can differ substantially, as feature reduction may involve the generation of meta-features from a previous dataset, while feature selection seeks to reduce a large dataset to an optimized subset of features. The point of both is to make the classification process simpler, faster, and more accurate by reducing dataset complexity. Feature reduction algorithms include ICA, PCA, common spatial patterns (CSP), and genetic algorithm (GA) feature selection. Each is explained in greater detail in the following sections. CSP and GAs have been utilized in BCI successfully, warranting further investigation (Zhang et al., 2010). However, issues arose during the implementation of each feature. A new algorithm, average distance feature selection, was later added to the list.

4.2.1 Independent Component Analysis

Independent Component Analysis (ICA) is a technique for separating a signal into more statistically independent subcomponents, each providing a unique contribution to the signal. The two assumptions used in computation of unique components are that the sources are separate and possess a non-Gaussian distribution. A weights matrix of coefficients is calculated to more effectively separate the sources, but the task is computationally intensive. Due to the resources required, standard ICA algorithms are unable to operate in real time, although ICA-preprocessed data can be used to train a classifier.

ICA Calculation

1) Take matrix **X**, with dimensions EEG channels *p* by samples *n*.
2) Perform mean removal in **X**.
3) Calculate whitening matrix **W** for **X**.
4) Select and reject specific ICs based upon results.
5) Finish with ICA processed EEG data **Y**.

In EEG signal processing, ICA has been used to eliminate eye blinks and other artefacts in preprocessing before feature extraction (Peiris et al., 2011). The dataset used in the benchmark microsleep classification results used ICA to remove ocular artefacts in the training and the testing, in conjunction with artefact pruning. In contrast, results close to the microsleep classification results, including a phi of 0.38, were achieved without ICA preprocessing (Davidson et al., 2007). The inclusion or absence of ICA resulted in the generation of EEG feature sets without ICA preprocessing for a more thorough analysis.

4.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is a commonly used dimensionality reduction technique. It comprises an orthogonal transform applied to the input observations. The transformation results in a matrix of "principal components," arranged in order of effect on data variability. The first principal component corresponds to the variable with the largest variability, and so on. The PCA transformation is orthogonal, rather than select collinear features. Also, it generates meta-features rather than selecting existing features.

PCA Calculation

1) Take training data matrix **X** with dimensions features p by observations n.
2) Perform mean removal in **X**.
3) Calculate covariance matrix **R** for **X**.
4) Perform eigenvalue decomposition of **R**.
5) Select PCs based upon results.
6) Take projection matrix **D**, apply it to training data, and apply it to testing data.
7) Reduce sizes of training and testing matrices to desired number of features.

PCA is an unsupervised method of feature reduction and, as a result, it does not require *a priori* knowledge of class labels. While it was used previously (Peiris et al., 2011), the possibility remains that a supervised feature selection method may yield better results (Omary, 2009). PCA has many variations, but the standard algorithm was used as to serve as a baseline comparison to other methods. An adjustable cap of meta-features was inserted, as to capture the ones most responsible for the high variance.

4.2.3 Common Spatial Patterns

Common Spatial Patterns (CSP) is a dimensionality reduction technique related to PCA. CSP has previously been utilized in BCI and biosignal processing (Zhang et al., 2010). CSP changes the variance between two particular classes, one with maximized variance and one with minimized variance. PCA is a method of unsupervised learning, while CSP is a method of supervised learning since CSP requires examples of each class to form the spatial filter. CSP can have better performance than PCA, and creates new features like PCA (Lu et al., 2009).

CSP Calculation

1) Take training data matrix **X**, with dimensions features p by observations n.
2) Perform mean removal in **X**.
3) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
4) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
5) Compute correlation matrices $\mathbf{R_n}$ and $\mathbf{R_e}$ from $\mathbf{X_n}$ and $\mathbf{X_e}$ respectively.
6) Solve eigenvalue decomposition of both $\mathbf{R_n}$ and $\mathbf{R_e}$.
7) Sort eigenvalues in descending order.
8) The highest eigenvalue corresponds to maximized variance, and lowest corresponds to minimized variance.
9) Select additional eigenvalues based on the number of features.
10) Apply transformation matrix **W** to training data and to testing data.

The changes in variance have made class differences more apparent in BCI and signal processing (Yin et al., 2008). CSP can be applied in the temporal domain as well as the frequency domain, as it is a linear transform. CSP performance may drop if artefacts are used to train it. Different variations of CSP can be applied to lapse detection, including: traditional CSP (Yin et al., 2008), regularized CSP (RCSP) (Lu et al., 2009), mixtures of CSP (MCSP) (Sun et al., 2008), and others (Zhang et al., 2010). As such, CSP was investigated.

4.2.4 Projection to Lateral Subspaces

Projection to Lateral Subspaces (PLS), also known as partial least squares regression, is a supervised feature reduction technique based upon regressive linear modelling. Like PCA, it is an orthogonal technique rather than one dependent upon highly collinear features. It also generates a transformation matrix that is based upon the calculation of a covariance matrix. Unlike PCA, PLS uses class label information (Chen, 2005; Muradore, 2012).

PLS Calculation

1) Take training data matrix **X** with dimensions features p by observations n.
2) Take training data class labels vector t of length n.
3) Perform mean removal in **X**.
4) Calculate covariance matrix **R** for **X**.
5) Using class labels matrix t, Perform partial least squares regression to find transformation matrix **D**.
6) Apply **D** to training and testing data.
7) Reduce sizes of training and testing matrices to desired number of features.

PLS has multiple variations and has been successfully used in EEG classification of evoked potentials (Chen, 2013; Hutapea, 2014).

4.2.5 Genetic Algorithm Feature Selection

Genetic algorithm feature selection is the optimization of feature combinations based on successful classification results (Kim et al., 2006; Parini et al., 2007; Wang et al., 2011). The function mimics the process of natural selection across generations. A genetic representation of a solution and a fitness function to optimize solutions are both required in a genetic algorithm. With each iteration, randomized collections of features are generated, loosely analogous to producing offspring. Based on the classification results, the fitness of each feature collection is selected to produce the next generation. This process optimizes feature selection fitness over generations, and could be applied to microsleep detection.

The criteria used to define fitness can vary greatly, ranging from performance on a classifier to regression models to effect size (Kim et al., 2006; Parini et al., 2007; Wang et al., 2011). The decision was made to implement GAs as a feature selection method for microsleep detection. Unlike PCA, the preferred implementation of GAs generated groups of features, rather than new features. The feature groups selected by GAs corresponded to specific spectral features on electrode channels, so they could be used to directly reduce information to the most relevant for microsleep detection. In context of EEG-based

microsleep detection, "collinear" features were those reflective of the same spectral changes across multiple channels. Group selection of features was random at first, so as to ensure collinear features were not selected.

Time is required to run and optimize GAs, but an optimal feature set can be reused extensively once calculated. Signs of microsleeps in different individuals can vary widely. The primary reason for using genetic algorithms in microsleep detection is that a GA-based system may find a combination of features uniquely suited and optimized for each subject configuration.

GA Calculation

1) Take training data matrix **X** with dimensions features p by observations n.
2) Select a random subset of features in **X**.
3) Estimate fitness of each feature subset based upon a specific criterion.
4) Compute feature subset corresponding to highest fitness is used as basis for other subsets.
5) Repeat (3) and (4) until error goal or number of iterations met.
6) Reduce training data **X** to feature subset corresponding to highest fitness is retained.
7) Reduce testing data to same feature subset.

Time requirements for GAs are a drawback, but feature sets can be generated for later usage. Once features have been defined, GAs are no longer needed for real-time operation. GAs were implemented. From a prototype of GA, a new method of feature selection was developed.

4.2.6 Average Distance Feature Selection

Average distance feature selection was originally implemented as a component of GA. It was intended as a method of measuring the fitness of individual features. All features corresponding to each class (events and non-events) are averaged together across all observations. The resultant vectors are then subtracted from each other, and the largest distances found. The features corresponding to the largest average distances are retained, and the rest discarded. The number of features can be adjusted. The process is known as the Average Distance between Events and Non-events (ADEN). Many tests were performed

utilizing ADEN on artificial data, and a handful of tests investigated increasing the number of features beyond one with ADEN, as shown in the Appendix. Four variations of ADEN were developed.

4.2.6.1 ADEN

ADEN required the user to define U features to retain. The training data $\mathbf{X}$ consisted of F features and M observations. Then, features corresponding to events and nonevents were separated into $\mathbf{X}_{\mathrm{e}}$ and $\mathbf{X}_{\mathrm{n}}$. Each was averaged to form a mean feature vector (F long), $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$. The difference formed a single vector, $\Delta\overline{x}_f$.

$$\Delta\overline{x}_f = abs(\overline{x}_{\mathrm{e},f} - \overline{x}_{\mathrm{n},f}) \qquad (6)$$

The difference between classes was normalized by dividing vector $\Delta\overline{x}_f$ by Cohen's d (effect size), such that within-group variances in the training data could be accounted for. Training data $\mathbf{X}$ were reduced to a matrix of U features and M observations, with all remaining features based on the indices f of the U terms in $\Delta\overline{x}_f$. The testing data would likewise be reduced to u features, selected from the f indices corresponding to features in the training data.

ADEN Calculation

1) Take training data matrix $\mathbf{X}$, with dimensions features F by observations M.
2) Calculate Cohen's *d*.
3) Move all observations of non-events from $\mathbf{X}$ into matrix $\mathbf{X_n}$.
4) Move all observations of events from $\mathbf{X}$ into matrix $\mathbf{X_e}$.
5) Average $\mathbf{X_e}$ and $\mathbf{X_n}$ to form a mean feature vector (F long), $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$.
6) Calculate the absolute value of the difference between $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$ in vector $\Delta\overline{x}_f$.
7) Divide vector $\Delta\overline{x}_f$ by Cohen's *d*.
8) Arrange values in vector $\Delta\overline{x}_f$ in descending order.
9) Reduce training data $\mathbf{X}$ to features corresponding to U highest differences for training data.
10) Reduce the testing data to the same feature subset.

4.2.6.2 ADENZ

A second variation of ADEN was dubbed Average Distance Between Events and Non-events by Z-score transform (ADENZ). The *z*-score transformation involved subtraction of the mean for each variable, followed by dividing by the variable's standard deviation. In contrast with ADEN, ADENZ applied independent *z*-score transformations to the training and testing data, omitting Cohen's *d* (effect size).

ADENZ Calculation

1) Take training data matrix **X**, with dimensions features F by observations M.
2) Perform *z*-score transformation on **X**.
3) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
4) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
5) Average $\mathbf{X_e}$ and $\mathbf{X_n}$ to form a mean feature vector (F long), $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$.
6) Calculate the absolute value of the difference between $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$ in vector $\Delta\overline{x}_f$.
7) Arrange values in vector $\Delta\overline{x}_f$ in descending order.
8) Reduce training data **X** to features corresponding to U highest differences for training data.
9) Reduce the testing data to the same feature subset.

4.2.6.3 GADEN

A further development of ADEN was the incorporation of aspects of GA, resulting in Genetic Average Distance between Events and Non-events (GADEN). ADEN's primary role in GADEN was as a bottleneck for ranked features, as GA would be performed upon random combinations of remaining, selected ADENs. The user was required to designate a pool of V features to select as a bottleneck. A total of U features would be selected at random from the "gene pool" of Vfeatures. Approximately half of the training data would be randomly selected, and tested on the other half using only the selected U features.

GADEN Calculation

1) Take training data matrix **X**, with dimensions features F by observations M.
2) Calculate Cohen's *d*.
3) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
4) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
5) Average $\mathbf{X_e}$ and $\mathbf{X_n}$ to form a mean feature vector (F long), $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$.
6) Calculate the absolute value of the difference between $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$ in vector $\Delta\overline{x}_f$.
7) Divide vector $\Delta\overline{x}_f$ by Cohen's *d*.
8) Arrange values in vector $\Delta\overline{x}_f$ in descending order.
9) Reduce the training data **X** to the features corresponding to the V highest differences for training data.
10) Select a random subset of U features in **X**.
11) Estimate the "fitness" of each feature subset from phi correlation of training and testing on only each U-sized feature subset with LDA.
12) Use the feature subset corresponding to highest phi correlation as the basis for other subsets.
13) Repeat (3) through (12) until error goal or number of iterations met.
14) Reduce the training data **X** to retain only the feature subset of size U corresponding to the highest "fitness."
15) Reduce the testing data to same feature subset.

The random combination of features with the highest phi correlation would "reproduce" a new set of variants (e.g., new "genotypes" of a new "generation") that would be tested against the parent. The standard configuration of GADEN utilized a total of three generations with a constant number of "offspring" for each generation. GADEN took much more time and processing power than ADEN or ADENZ, but was able to overcome the potential issue of selecting collinear features present in the other two methods.

4.2.6.4 GADENZ

A further version of GADEN was developed, based upon ADENZ.

GADENZ Calculation

1) Take training data matrix **X**, with dimensions features F by observations M.
2) Perform *z*-score transformation on **X**.
3) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
4) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
5) Average $\mathbf{X_e}$ and $\mathbf{X_n}$ to form a mean feature vector (F long), $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$.
6) Calculate the absolute value of the difference between $\overline{x}_{\mathrm{e}}$ and $\overline{x}_{\mathrm{n}}$ in vector $\Delta\overline{x}_f$.
7) Arrange values in vector $\Delta\overline{x}_f$ in descending order.
8) Reduce the training data **X** to features corresponding to V highest differences for training data.
9) Select a random subset of U features in **X**.
10) Estimate the "fitness" of each feature subset from phi correlation of training and testing on only each U-sized feature subset with LDA.
11) Use the feature subset corresponding to highest phi correlation as the basis for other subsets.
12) Repeat (3) and (4) until error goal or number of iterations met.
13) Reduce the training data **X** to retain only the feature subset of size U corresponding to the highest "fitness."
14) Reduce the testing data to same feature subset.

The primary difference with GADENZ was the use of a *z*-score transform to normalize the training data. Cohen's *d* was not used. The different method of normalization was thought to potentially result in a different set of features than GADEN.

4.3 Pattern Recognition

Pattern recognition is the automatic sorting of data into categories or assigning labels. Pattern recognition techniques are often grouped by the learning technique utilized. A

successful pattern recognition algorithm can correctly identify the label of testing data the majority of the time. Two common categories are supervised and unsupervised learning. Supervised learning involves a classifier being given a labelled set of training data, and generalizing each group. Unsupervised learning does not contain labels, and is focused on discerning patterns between groups, irrespective of class labels.

Three supervised learning approaches to pattern recognition were investigated: linear discriminant analysis (LDA), support vector machines (SVMs), and radial basis functions (RBFs).

4.3.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a simple type of pattern classification algorithm. LDA calculates the within-group and between-group variances of training data and draws a boundary between them. Depending on which side of the boundary that a new observation is assigned to, it is assigned a different group label. LDA was used previously in microsleep detection (Peiris et al., 2011) and has the advantages of being a robust and simple classifier.

LDA Calculation

1) Take the matrix of training data **X** and target vector $\overline{x}_{\mathrm{t}}$.
2) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
3) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
4) Calculate group mean and variance for $\mathbf{X_e}$ and $\mathbf{X_n}$.
5) Compute class separation to set threshold.
6) Expose the classifier to testing data.

LDA was the baseline that other pattern recognition algorithms were compared with.

4.3.2 Support Vector Machines

Support Vector Machines (SVMs) are a method of supervised learning based on the projection of a hyperplane into high dimensionality space (Ruping, 2001). The position of a data point relative to the hyperplane is used for classification. The algorithm positions the hyperplane utilizing the maximum distance from each group of training instances. Once an

optimal position for the plane has been found, all other instances are discarded except for the "support vectors." SVMs are often computationally complex but are usually accurate on training data (Ruping, 2001). SVMs can turn high dimensionality problems into linear classification problems. SVMs have previously been successfully used in lapse detection (Golz et al., 2007; Krajewski et al., 2008; Golz and Sommer, 2010), but there are other potential kernels (Qiao et al., 2010; Xu et al., 2010) that can be explored. SVMs are complex to implement but once the support vectors have been found, classification becomes a linear process (Deng, 2011). SVMs, given prior performance, were recommended to be used in microsleep detection. Specifically, two SVM kernels were successfully used on dummy data, the Gaussian kernel and polynomial kernel. The SVM Gaussian (SVMG) kernel operates similar to a radial basis function (RBF).

SVMG Calculation

1) Take the matrix of training data **X** and target vector $\overline{x}_{\mathrm{t}}$.
2) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
3) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
4) Assuming a Gaussian distribution of data, fit a hyperplane that maximizes distance between features of $\mathbf{X_e}$ and $\mathbf{X_n}$.
5) Compute class separation to set threshold.
6) Expose the classifier to testing data.

The SVM polynomial (SVMP) kernel fits a higher order model to training data.

SVMP Calculation

1) Take the matrix of training data **X** and target vector $\overline{x}_{\mathrm{t}}$.
2) Move all observations of non-events from **X** into matrix $\mathbf{X_n}$.
3) Move all observations of events from **X** into matrix $\mathbf{X_e}$.
4) By adjusting a polynomial model, fit a hyperplane that maximizes distance between features of $\mathbf{X_e}$ and $\mathbf{X_n}$.
5) Expose the classifier to testing data.

Both were compared against each other in terms of performance to determine if one was superior for microsleep detection.

4.3.3 Radial Basis Functions

Radial basis functions (RBFs) have been utilized alongside neural networks in machine learning research and can be used in unsupervised or supervised learning (Xu et al., 2004; Xu et al., 2010). RBFs had previous usage in EEG classification (Xu et al., 2004), but not microsleep detection. RBFs have been used in BCI successfully, so they may be applicable to microsleep detection (Bassani and Nievola, 2008). An RBF network lacks the back-propagation of a neural network, and can be used in place of a traditional neural network (Xu et al., 2004). When used as a neural network, radial basis functions compute weights with the "*k* nearest neighbor" algorithm to calculate an output. An RBF classifier has not yet been utilized in microsleep detection research, but a related fuzzy-logic-based classifier has been used with spectral features in vigilance tests (Coufal, 2009). Previous research in machine learning has utilized neural networks and compared them with RBFs (Chundi et al., 2004), but this has not yet been performed in microsleep detection. Under certain circumstances, RBFs can perform better than neural networks, such as in speaker recognition (Finan et al., 1996), or at least perform comparably under other circumstances.

RBF Calculation

1) Take the matrix of training data **X** and target vector $\overline{x}_{\mathrm{t}}$.
2) Initialize neurons with random weights.
3) Use training data **X** and target vector $\overline{x}_{\mathrm{t}}$ to adjust neuron weights, clustering the datapoints.
4) Iterate until the error goal reached.
5) Expose the classifier to testing data.

RBFs have not yet been utilized in microsleep detection, but the success of a fuzzy-logic-based classifier with RBF success in other areas made them attractive for use in EEG-based microsleep detection (Finan et al., 1996; Coufal, 2009).

4.4 Classifier Structure

The structure of a classification system can be altered radically. Performance can drastically change when algorithms are changed, or when training scenarios differ. Ensembles of classifiers can have significant improvements over a single classifier, but typically require more resources to run. Single classifier structures, in addition to four types of ensembles, were evaluated.

4.4.1 Single Classifier

A single classifier is trained on data from training subjects, and then evaluated on the testing subject. The task is repeated until each subject had been utilized as the testing subject. Performance metrics are were averaged together.

Single Classifier Usage

1) Set aside one subject for testing, and use the rest for training.
2) Train one classifier for each training subject.
3) Evaluate all classifiers on a test subject.
4) Average all classifier outputs together into a single output vector.
5) Calculate performance metrics.
6) Change the subject used for testing.
7) Repeat steps (1) to (6) until all subjects are tested.
8) Average all performance metrics together.

While ensembles may offer theoretical benefits, single classifier LOOCV was previously used as a baseline to compare to an ensemble (Peiris et al., 2011). Four ensembles of classifiers were considered for use in microsleep detection: bagging, boosting, AdaBoost, and stacking. A previous study utilized a classifier based on stacked generalization (Peiris et al., 2011).

4.4.2 Bagging

Bagging was the first of four methods considered which employ an ensemble of classifiers. Unweighted majority voting (based on “bagging”) (Breiman, 1996), with each

classifier used to determine which state a given instance belongs to, means each classifier's vote is equally weighted (Alpaydin, 1992).

Bagging Ensemble Usage

1) Set aside one subject for testing, and use the rest for training.
2) Take all training data and randomly reorder it.
3) Divide reorganized training data into a number of blocks.
4) Train one classifier for each training data block.
5) Evaluate all classifiers on test subject.
6) Average all classifier outputs together into a single output vector.
7) Calculate performance metrics.
8) Change the subject used for testing.
9) Repeat steps (1) to (8) until all subjects are tested.
10) Average all performance metrics together.

Bagging was implemented so that the number of randomized blocks could be adjusted.

4.4.3 Boosting

Boosting gives more influence to more successful classifiers (Schapire et al., 2005). Boosting consists of a majority vote on three classifiers. The first classifier is trained on a subset of training data. The second classifier is trained on a portion of the data correctly and incorrectly classified by the first. The third classifier is used on observations where the first two classifiers disagreed.

Boosting Ensemble Usage

1) Set aside one subject for testing, and use the rest for training.
2) Train one classifier using a random subset of training data including both classes.
3) Train a second classifier, trained on portions of the subset correctly classified data and incorrectly classified data from the first classifier.
4) Train a third classifier on the remainder of the training data for instances when the first two disagree.
5) Evaluate the classifier ensemble on test subject.
6) Average all classifier outputs together into a single output vector.
7) Calculate performance metrics.
8) Change the subject used for testing.
9) Repeat steps (1) to (8) until all subjects are tested.
10) Average all performance metrics together.

Boosting was implemented.

4.4.4 AdaBoost

Adaptive boosting (AdaBoost) is an ensemble of weak learners, or simplified linear classifiers. Each classifier prioritizes the correct classification over observations that previous weak learners could not successfully classify (Freund and Schapire, 1997).

AdaBoost Ensemble Usage

1) Determine number n of weak learners.
2) Set aside one subject for testing, and use the rest for training.
3) Train one classifier, a weak learner, on a random subset of training data including both classes.
4) Increase weights on incorrectly classified datapoints.
5) Weight the weak learner based on correctly classified datapoints.
6) Have the next weak learner attempt to correctly classify datapoints with highest weights.
7) Repeat steps (3) to (6) until n weak learners generated.
8) Evaluate classifier ensemble on the test subject.
9) Calculate performance metrics.
10) Change the subject used for testing.
11) Repeat steps (1) to (10) until all subjects are tested.
12) Average all performance metrics together.

While stacking and bagging modules were implemented using several different pattern recognition modules, AdaBoost was only implemented with LDA due to the requirement for a weak learner.

4.4.5 Stacking

Stacked generalization (or "stacking") aims to combine the individual classifiers with a meta-learner (Gandhi et al., 2006). A portion of the training data is held back to form a "pseudo-testing" dataset. A linear model is fitted to the ensemble's performance on the pseudo-testing data, helping to generate the meta-learner. Afterwards, the meta-learner is presented the testing data (Wolpert, 1992).

Stacking Ensemble Usage

1) Set aside one subject for testing, and use the rest for training.
2) Set aside one training subject as pseudo-testing subject.
3) Train one classifier for each training subject.
4) Evaluate all classifiers on pseudo-testing subject.
5) Based on outputs from each classifier, use linear regression to weight each individual classifier.
6) Repeat (2) to (5) until all subjects used as pseudo-testing subject.
7) Evaluate each configuration on the testing subject.
8) Calculate performance metrics.
9) Change the subject used for testing.
10) Repeat steps (1) to (9) until each subject has been used as the testing subject.
11) Average all performance metrics together.

Stacking has previously been applied to the microsleep detection problem, setting the current performance benchmark (Peiris et al., 2011).

4.5 Technique Summary

Some of the listed techniques have been used in prior work. Log spectral power was used previously as a feature extraction method. For FS/R, PCA and ICA were used in the prior work. For classification, LDA was used in both a single classifier and in a stacking ensemble (Peiris et al., 2011). Linear spectral power was not used for feature extraction in the prior benchmark. The ADEN variants, PLS, and CSP were not utilized for FS/R. In addition, RBFs, SVMs, bagging, boosting, and AdaBoost were not used for classification. Before being used on EEG data, the implemented modules were validated.

CHAPTER 5. DATASETS

5.1 Introduction

Three separate datasets were examined using the microsleep detection software. The first dataset comprised simulated events, 2.0-s bursts of 15 Hz sinusoid, superimposed on 5 min of 16-channel EEG, with the signal-to-noise ratio (SNR) varied. The simulated dataset was utilized to validate the software implementation of prior modules and evaluate their limitations. After this, EEG data with expert-rated microsleep events from Study A (N = 8) and Study C (N = 10) were examined. Due to the breadth of the parameters examined, different permutations of each dataset were developed for comparison.

5.2 Artificial Event Dataset

After debugging ICTOMI modules, questions remained regarding their potential performance on detecting microsleeps in the Study A dataset. The occurrence of microsleeps is rare relative to non-events, forming a highly imbalanced dataset. To approximate Study A, an artificial "gold standard" dataset was programmed, with an artificial event superimposed on a subset of the Study A data. Different parameters of the artificial dataset were varied, such as the ratio of events to non-events and the signal-to-noise ratio. The purpose of this testing was to confirm that ICTOMI was working correctly on a dataset of precisely known events and to determine how signal power and class balances affect performance.

The artificial data were generated to loosely approximate the microsleep detection task. However, the advantage of an artificial dataset was the ability to exactly control the parameters of the event to be detected. The event for the artificial dataset was a 15 Hz sine wave, lasting for a total duration of 2 s. Five minutes of 16-channel EEG data were taken from each subject in the Study A dataset and further subdivided into 2-s segments, each with 50% overlap with the prior segment. A total of six segments had the sine wave added to all channels of the 16-channel data, resulting in 98% of the time being non-events and 2% being events. A total of 34 EEG band-derived spectral features were then taken from each segment of each channel, resulting in 544 features for 300 segments for each subject.

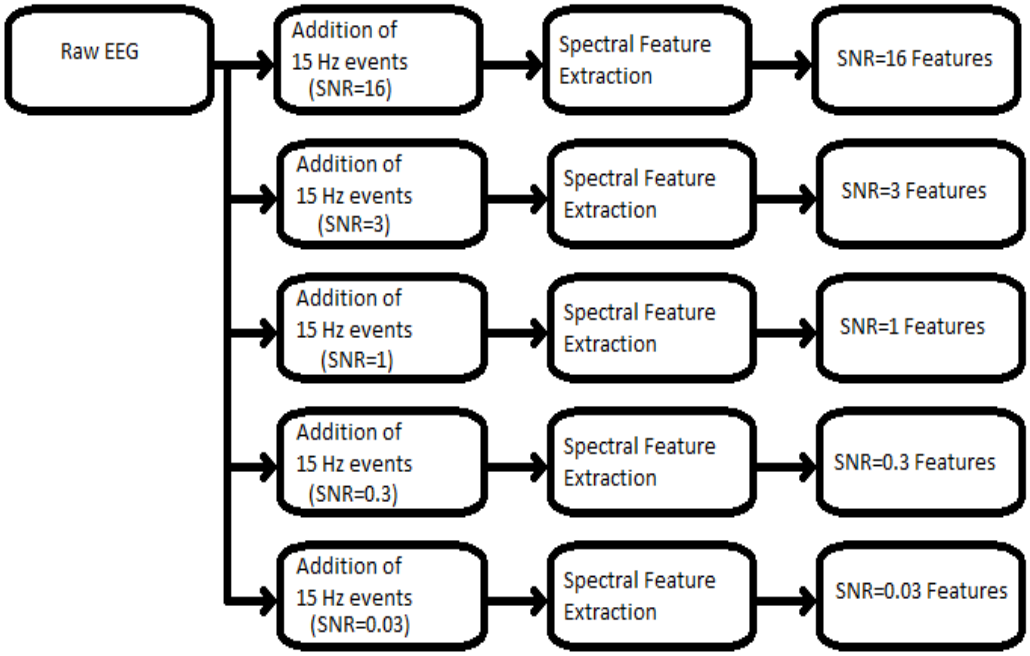

Figure 5.1: Generation schematic for artificial event features

The amplitude of the sine wave was scaled relative to the maximum EEG signal amplitude. The sine wave amplitude was scaled to a signal-to-noise ratio (SNR) of 16, 3, 1, 0.3, and 0.03 based on the amplitude rather than power. The root mean square amplitude of the signal is A_{signal}, while the amplitude of the noise is A_{noise}.

$$SNR = (\frac{A_{signal}}{A_{noise}})^2 \qquad (7)$$

Examples of an event are shown in Figures 5.2-6. The event that can easily be identified visually is shown in Figs. 5.2 and 5.3.

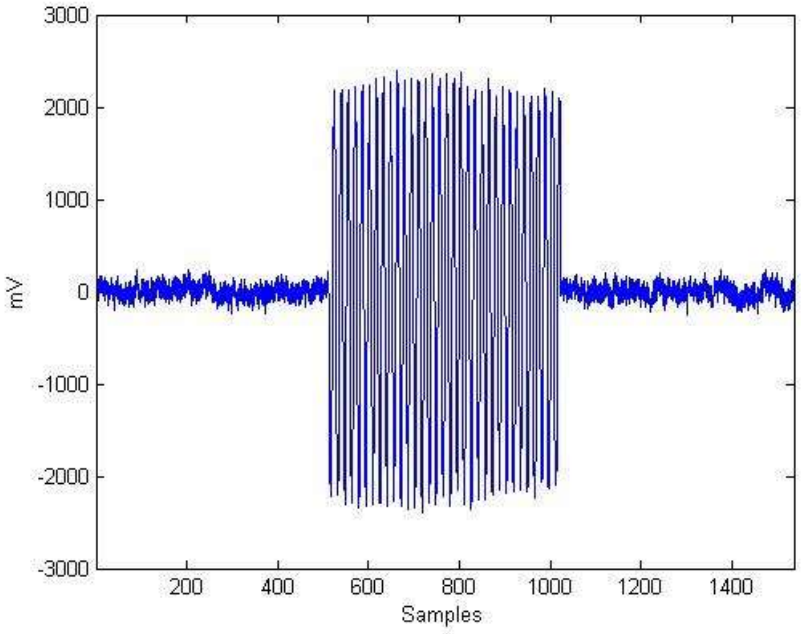

Figure 5.2: A combination of EEG and very easy event (SNR = 16.0)

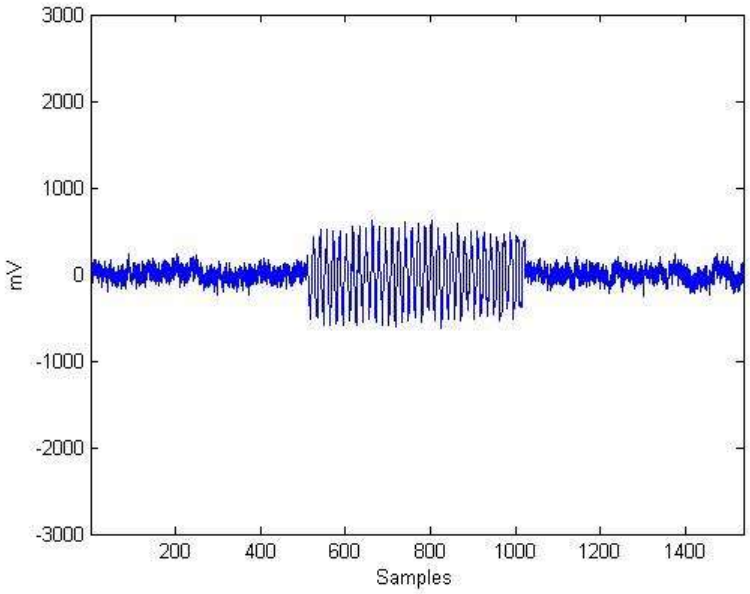

Figure 5.3: A combination of EEG and easy event (SNR = 3.0)

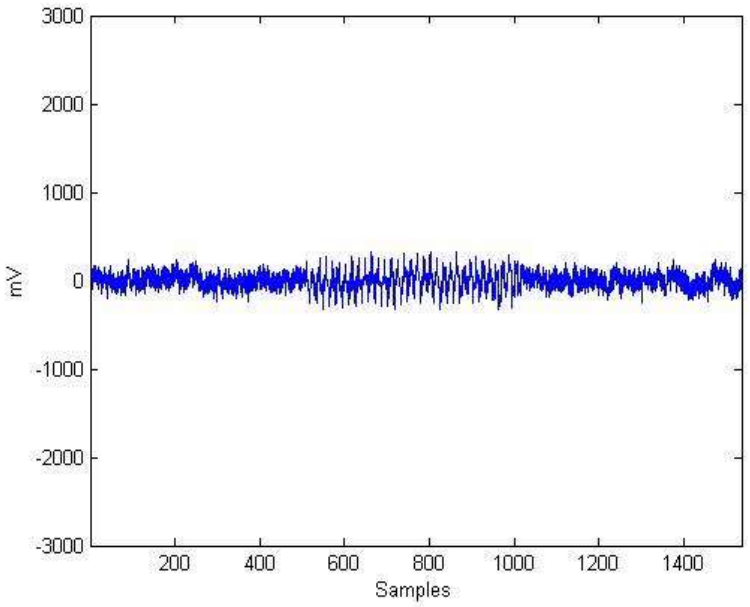

Figure 5.4: A combination of EEG and medium event (SNR = 1.0)

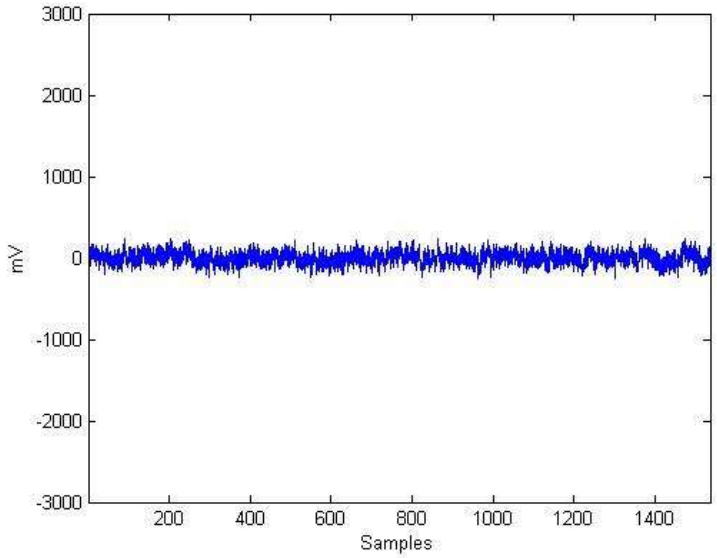

Figure 5.5: A combination of EEG and hard event (SNR = 0.3)

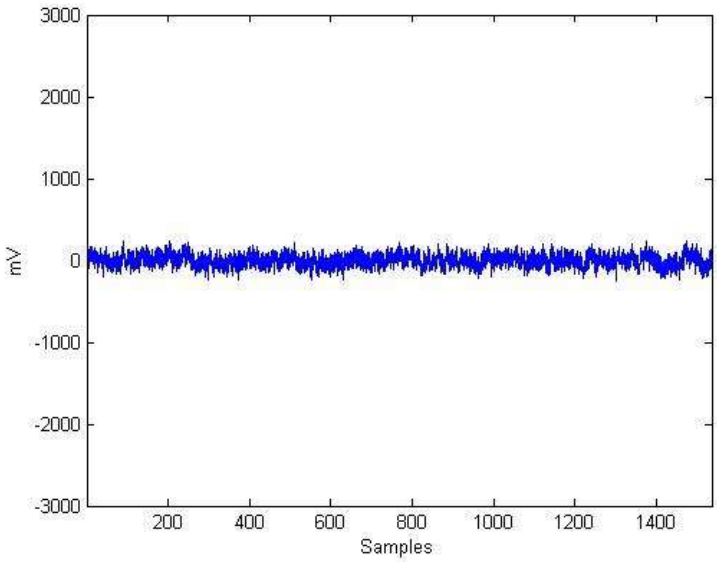

Figure 5.6: A combination of EEG and very hard event (SNR = 0.03)

Five other artificial datasets were generated, identical to the previously described ones, except with equal numbers of events and non-events. Class balance was achieved by repeating events and randomly deleting a subset of non-events until the ratio of events to non-events was unity. All datasets were kept independent of each other, with each having 6 artificial events superimposed over 5 min of EEG data.

5.3 Study A

The first EEG dataset to be used in research is the Study A dataset (N = 8). The Study A dataset was recorded at 256 Hz, using 16 referential channels (Peiris et al., 2011). Two non-consecutive sessions, each approximately one hour in length, were recorded for each subject.

Notch filtering was used to remove 50 Hz power interference. The dataset has been extensively used in prior research, establishing the current baseline for lapse detector performance (Davidson et al., 2007; Peiris et al., 2011).

5.3.1 Study A Gold Standard

Validation of training and testing data required properly-labelled states indicating a microsleep. The presence of a microsleep was treated as a binary state, where "1" indicated the presence of a microsleep and "0" indicated the responsive or baseline state. Data rated by human experts served as the gold standard for gauging performance of an automated classifier. Performance on a 1-D tracking task and video recordings were used to estimate the behavioural gold standard of alertness.

The rated data was integral to an automated lapse detection system (Peiris et al., 2011), acting as the "gold standard" for determining if an EEG segment was a microsleep or not. Previous datasets, such as the Study A, were rated by experts using both tracking and video data. The Study A utilized a 6-point scale to measure alertness: 1 = alert, 2 = distracted, 3 = forced eye closure while alert, 4 = light drowsy, 5 = deep drowsy, and 6 = sleep (including microsleeps). "Alert" periods were identified by fast eye blinks and normal facial tone. "Distracted" intervals included momentary diversions from the task. "Forced eye closure" was an instance of the subject closing their eyes while remaining alert. The "light drowsy" state was characterized by the subject's blink rate slowing and facial tone shifting. The "deep drowsy" state had the subject show fewer eye movements and partial eye closure. The 'sleep' state had prolonged eye-lid closure with head nodding and jerks. Values on the scale from 6 counted as lapses.

CTT performance data was rated independently from the video data. A "flat spot" was a location in the tracking data where the subject stopped responding, lasting at least 300 ms. Flat spots with longer durations were examined for overlaps with video microsleeps. Definite microsleeps were considered combinations of flat spots and video lapses.

While rating was performed conservatively, some segments were ambiguous as to whether they were microsleeps or not. Only definite microsleeps and segments rated 6 on the scale by Peiris were used as the gold standard. However, non-microsleep lapses were utilized or removed under various training scenarios.

If a microsleep or flat spot occurred anywhere within a 2-s window corresponding to a segment of EEG, the entire window was marked as an event for Study A. While the gold standard was not a perfect measure of the brain-state during microsleep events, it was as close

as the human experts could provide. Different gold standards were developed, corresponding to different scenarios for defining events:

1) Definite microsleeps (simultaneous video microsleeps and flat spots) = 1, and all other states = 0.
2) Flat spots only = 1, and all other states = 0.
3) Video microsleeps only = 1, and all other states = 0.
4) All lapses (video microsleeps and/or flat spots) = 1, and all other states = 0, as per prior research (Davidson et al., 2007; Peiris et al., 2011).
5) Definite microsleeps (with simultaneous video microsleeps and flat spots) = 1, most other states = 0, apart from removal of segments of flat spots only and videos only from the analysis: *Pruning*.

Due to the imbalance of classes, "balanced" versions of the Study A and gold standards were also developed. To artificially balance the dataset, events were repeated and a random subset of non-events was deleted until the total composition of the dataset was evenly split between events and non-events. The possibility that a classifier trained on balanced data and tested on unbalanced data could perform better due to the removal of bias was considered and later evaluated.

5.3.2 Study A Preprocessing Reassessment

The Study A feature set used to achieve the current mean phi correlation benchmark of 0.39 (Peiris et al., 2011) had ICA performed on it bipolar EEG converted from referential EEG under the assumption that it would have less noise. However, a similar mean phi correlation of 0.38 was achieved on the same features without the use of ICA (Davidson et al., 2007). Both bipolar processed EEG and referential unprocessed EEG were examined to determine if bipolar features contained less noise. The gold standard used for the referential and bipolar EEG was the "lapse" event, so as to approximate previous work and to maximize the total number of events. The ICA-processed "clean" bipolar 544 spectral feature EEG, or Study A Bipolar ICA Spectral Power (SABIS), provided a valuable comparison against other feature set performances.

5.3.2.1 Alternative Feature Extraction

The other feature set developed from the ICA-preprocessed, artefact-pruned bipolar Study A EEG was the Study A feature set, which was known as the Study A Bipolar ICA Log Power (SABIL) feature set. Artefact pruning, as used in the SABIL and SABIS feature sets, was conducted by performing a *z*-score transform and rejecting any epoch greater than

30.0. While not elaborated by Peiris et al. (2011), the natural logarithmic transform of the entire power spectrum was used to calculate the spectral power of each band instead of the power spectrum estimate of the signal from the Burg algorithm (Peiris et al., 2011). Due to the lack of detail regarding differences in performance between spectral features and log spectral features, the initial tests were performed with both the SABIL and SABIS feature sets. Divergences in performance due to differing feature extraction methods were a gap requiring additional examination.

5.3.2.2 Raw Referential and Bipolar Feature Sets

The use of ICA eye blink removal and artefact pruning in the SABIL and SABIS feature sets removed an average of 578 epochs (208-1334) from each subject's total of 7200, resulting in a potential ~16% loss of information. Two other feature sets were reconstructed from the raw EEG from Study A. A completely new set of features were generated in each case, with 34 spectral features from each of 16 referential channels. A matrix of 544 features and 3600 observations was generated for each of the two 1-hour sessions per subject. In parallel with this, 16 bipolar channels used previously (Peiris et al., 2011) were calculated. The original was "raw referential," also known as the Study A Referential Unprocessed Spectral Power (SARUS) features. From the "raw bipolar" EEG features or the Study A Bipolar Unprocessed Spectral Power (SABUS) features, feature matrices of identical dimensions to the SARUS feature set were computed.

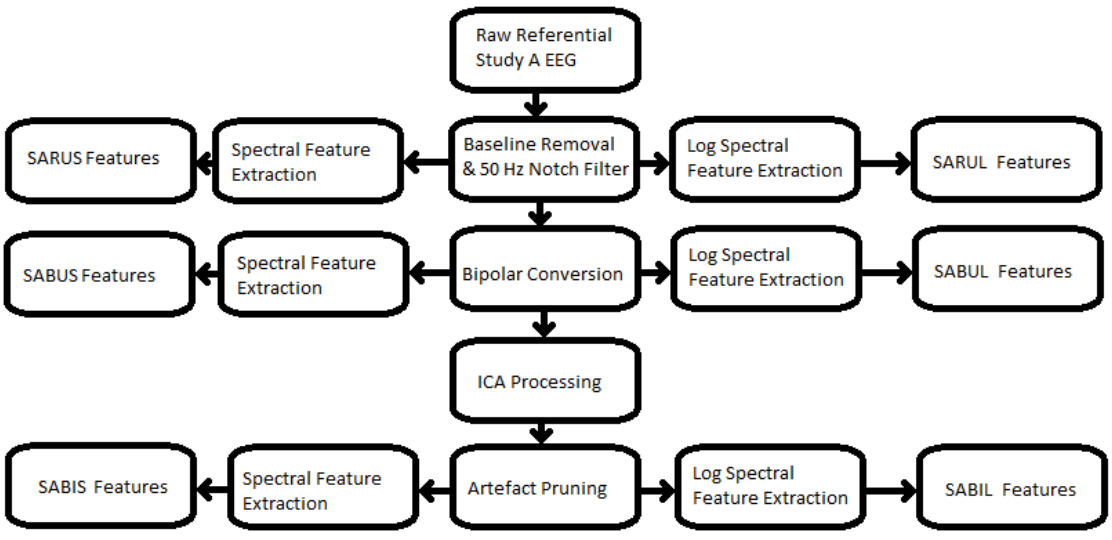

Figure 5.7: Study A feature set variations generation schematic

The SARUS and SABUS feature sets had both 1-hour sessions concatenated, resulting in a matrix of 544 features by 7200 observations per subject. Unlike prior work (Peiris et al.,

2011), no observations were deleted so as to better approximate a realistic scenario. The initial SARUS and SABUS feature sets were generated using the same feature extraction method as the SABIS features, linear spectral power. Variants of the SARUS and SABUS feature sets using the same log power feature extraction method as the SABIL features and additionally compared. The resulting variants were called the Study A Referential Unprocessed Log Power (SARUL) and Study A Bipolar Unprocessed Log Power (SABUL) feature sets.

5.4 Study C

Study C was examined in tandem with Study A. The Study C dataset was originally a combination of EEG and fMRI data, but only the EEG data were used to test automated microsleep detection. Originally, the study consisted of 20 subjects, but only 10 individuals with the largest number of microsleeps were analyzed further. The remaining Study C dataset (N = 10) used a 2D CTT in conjunction with video recording at 25 fps.

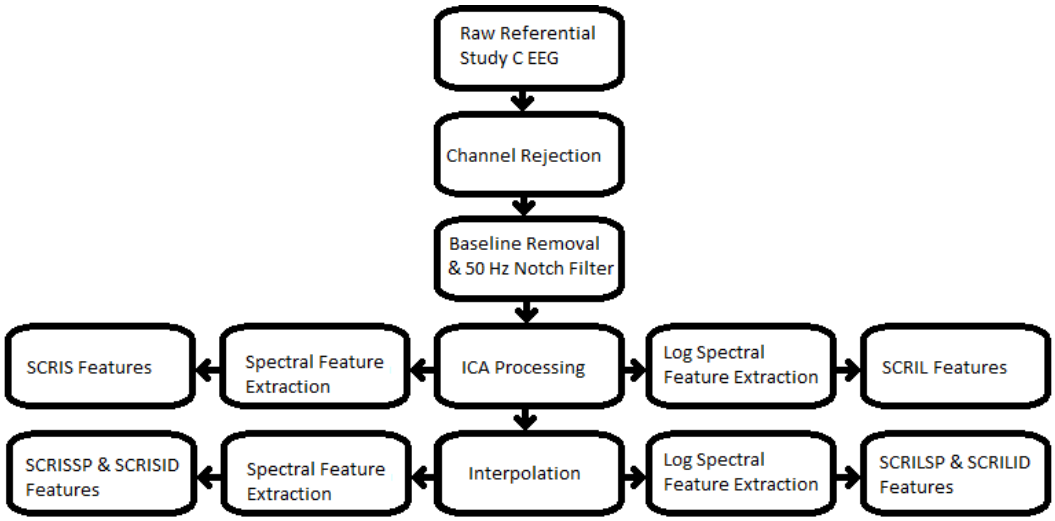

Figure 5.8: Study C feature generation schematic

Referential EEG from 64 channels was conducted, although several channels were discarded from each subject in preprocessing. The number of channels ranged from 30-60 per subject, with 17 channels consistent across all subjects. Additionally, ICA was performed to remove eye blink artefacts and overhead noise was filtered out. The primary feature set on Study C was a referential EEG dataset, Study C Referential ICA-Processed Spectral Features (SCRIS). Documentation of events, such as microsleeps, was also more meticulous (Poudel et al., 2010).

5.4.1 Study C Gold Standard

Data from Study C was examined utilizing similar formatting, feature extraction, feature selection/reduction, and classification to the Study A. Each observation was paired with a binary index indicating the presence or absence of an event. In the case of Study A, flat spots, video microsleeps, and definite microsleeps were all denoted as gold standard events. In the case of Study C, alert periods were treated as non-events, while events were defined as definite microsleeps and rest periods (sleep > 15 s) in the case of Study C.

Behavioural events in Study C were categorized as microsleeps, attention lapses, and impaired responsiveness events, rest periods and definite microsleep events as opposed to Study A's more ill-defined "lapses of responsiveness." In prior work (Peiris et al., 2011), the performance benchmark was performed on behavioural data where any video microsleep or flat spot was considered to be an event. The number of events in Study A was increased by using the "lapse" criterion rather than a "definite microsleep," which consisting of both video microsleeps and flat spots. The distinction between lapses of responsiveness and microsleeps was often in Study A compared to later work (Poudel et al., 2010). As such, Study C's gold standard was considered more reliable than Study A's gold standard.

5.5 Feature Extraction

5.5.1 Feature Details

In both Study A and Study C, 34 spectral features (described in Section 4.1) were calculated based upon a 2-s sliding window of EEG with a 50% overlap with the prior second. Three feature sets derived from Study A resulted in a feature matrix of 544 features for two hours of data per subject.

5.5.2 Study C Complications

The Study C data used was the same as Poudel et al. (2010), which had undergone substantial preprocessing due to having been recorded in an MRI scanner. Channels were rejected due to electrical impedances, ICA was performed to remove eye blinks, and filtering was done to remove overhead power. Additional preprocessing was required to remove the MRI gradient artefacts. Due to being far more preprocessed than Study A, a conscious decision made was to minimize changes to the Study C data and to simply test if the successful approaches from Study A could be directly applied to the Study C data.

Due to the uneven number of channels in Study C, null vectors were inserted to compensate for missing channels, resulting in 2040 features for 50 min of data per subject.

The usage of zero vectors, constant DC offsets, null events, and "Not a Number" (NaN) substitutions for missing features did not affect the results on intra-subject classification, but the zero vectors were included for simplicity. Interpolation using inverse distance and spherical modelling also failed to improve results. As a result, Study C was given a consistent number of features to enable inter-subject classification.

Potential improvements of taking the log of the power spectrum were investigated with Study C. The data was stored in referential format, and it was decided to not to convert to bipolar for Study C. Given the irregular numbers of channels between subjects and potential limitations of interpolation, a bipolar conversion of the Study C EEG added additional layers of complexity. As the decision had been made to limit additional preprocessing to Study C, the conversion to bipolar was not performed.

5.6 Planned Evaluations

5.6.1 Artificial Event Data Evaluation

The range of variations for the artificial event, Study A, and Study C feature sets presented a large range of possible tests to run, given the number of ICTOMI modules. As such, limiting the feature sets to a few informative ones rather than a larger number of less informative tests would allow more comprehensive testing to be utilized when required. For example, the artificial event datasets would be used to validate the basic ICTOMI modules so that the SNR of the artificial data could be compared with the performance of the EEG data. Additionally, the artificial data would be used on a set of basic approaches and then the primary research thrust would shift to the EEG datasets.

5.6.2 Study A Evaluation

5.6.2.1 Study A Clean and Expanded Data Evaluation

The Study A and Study C feature sets present far more variations than the artificial event datasets. For Study A, four primary feature sets were used. Two are "cleaned," meaning they are bipolar EEG with ICA and artefact pruning. The first SABIS feature set had spectral features, while the original SABIL one had features from the log of the power spectrum. The SABIL set was used in previous research (Peiris et al., 2011). However, it was uncertain if the SABIS feature set's spectral features contained enough information to successfully discern between classes.

5.6.2.2 Study A Raw Data Evaluation

The "raw" feature sets, SARUS and SABUS, lacked the artefact pruning and ICA filtering of the SABIS feature sets, and both used spectral features. The main purpose of comparing the SARUS and SABUS feature sets was to determine the differences in performance, if any, from using referential or bipolar EEG, a topic not covered in prior work (Peiris et al., 2011). The two feature sets additionally demonstrated the effects of leaving artefacts in and not performing ICA upon the EEG. Based upon prior work, the inclusion of ICA should not affect results, but feature extraction method and artefact pruning might (Davidson et al., 2007).

5.6.3 Study C Evaluation

The primary analysis on Study C had far less variants than Study A, due to a decision to reduce additional preprocessing on the data. The SCRIS features included "null" channels, or zeros inserted to allow for a common number of features.

Other variations, including with interpolation and calculating features by taking the log of the power spectrum, would be explored to see if they offered any changes or improvements in performance. The log features were referred to as Study C Referential ICA-Processed Log Spectral Features (SCRIL). Due to the primary data being stored in referential format, all variant Study C features were derived from referential data. Spherical coordinates and inverse distance interpolation were used with Study C with EEGLAB. The resulting features were referred to as SCRIS spherical (SCRISSP) and SCRIS inverse distance (SCRISID). The combination of interpolation with log spectral feature extraction resulted in the SCRIL spherical (SCRILSP) and SCRIL inverse distance (SCRILID) feature sets.

All were evaluated in a linear classifier, and unsuccessful variants were removed until only one feature set remained. While Study A had prior detection performance values in the literature (Peiris et al., 2011), Study C did not. If none of the variants performed successfully relative to the basic feature set, only the basic SCRIS feature set would be used in further research to establish a precedent. If SCRIS features proved sufficient for basic classification, they could form the basis for further work.

CHAPTER 6. SYSTEM EVALUATION ON SIMULATED EEG EVENTS

6.1 Introduction

Before the EEG-based feature sets were developed, the ICTOMI software toolset was validated with simulated EEG events. The process utilized performance metrics based upon prior research (Davidson et al., 2007; Peiris et al., 2011). The performance metrics of each classifier were averaged together. The performance metrics included: mean accuracy, specificity, sensitivity, phi, and area under the receiver operating characteristic (AUC-ROC) curve. Current performance metrics (Peiris et al., 2011) were utilized as a benchmark for comparison. The optimally performing configuration of algorithms and classifiers were used as the basis for an EEG-based microsleep detector. The phi correlation coefficient, however, quickly became the primary measure of classifier effectiveness, since it is independent of class distributions (Peiris et al., 2011).

The artificial datasets allowed for performance on known SNRs to be compared. The microsleep feature sets have a variable and unknown SNR exacerbated due to class imbalances. Comparing the results from an artificial event dataset with an EEG spectral feature set for a specific system configuration provided a method to estimate the SNR. A major contrast between the artificial event datasets and the rated EEG feature sets was that the target events in the artificial sets were consistent in duration, total number, and amplitude for all subjects, unlike the EEG feature sets. The artificial event datasets offered a view of uniform and consistent features across the same dataset.

Hypothesis 1: *Simulated EEG events with a variable SNR provide an estimate of performance on real EEG feature sets.*

Rationale: Each of the artificial datasets possessed a 15-Hz event that would appear on several of the spectral features used in the study. As the SNR drops, the mean phi performance will also drop.

6.2 Methods

All variants of the artificial datasets were subjected to the same battery of tests. The purpose of the initial tests was largely to ensure module arrangements functioned properly. Feature reduction modules included PCA, CSP, ADEN, and GADEN with 5 permutations over 3 iterations. Pattern recognition modules included LDA, RBF, SVM with a Gaussian kernel (SVMG), and SVM with a polynomial kernel (SVMP). Twelve primary configurations

were utilized, each a different arrangement of feature reduction and pattern recognition modules. Both balanced and unbalanced data were analysed, and the results from cross-validation are presented below. The phi correlation coefficient was utilized as the chief performance metric as it was found that the other metrics, such as accuracy, sensitivity, specificity, and selectivity, often varied greatly according to the ratio of non-events to events. A value of phi corresponding to "1" indicates perfect performance in successfully identifying all events and non-events. It was hypothesized that mean phi performance would drop as the SNR did for all configurations, with ensembles out-performing single classifiers.

6.3 Results

6.3.1 Single Classifier Performance

Single classifiers were tested prior to ensembles. In addition to phi, measures of sensitivity, specificity, and selectivity were calculated.

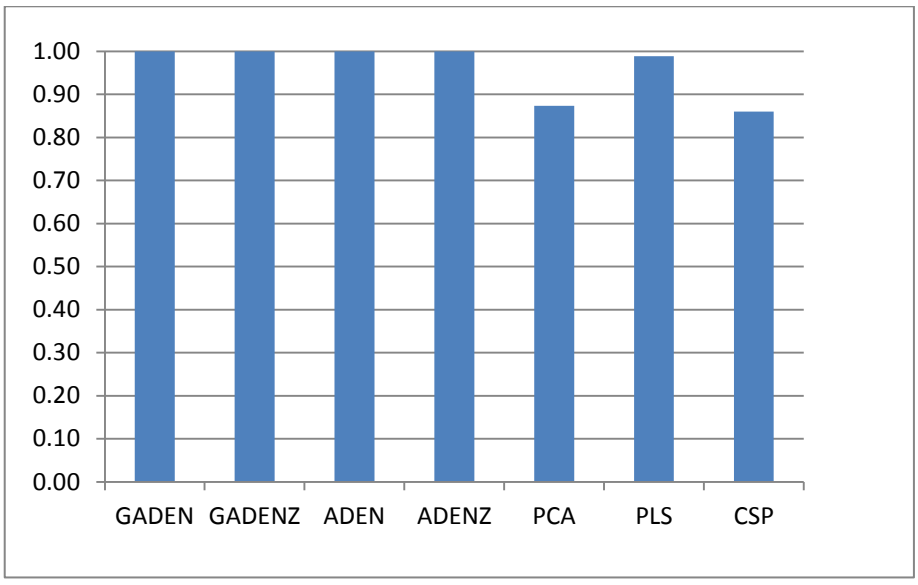

Figure 6.1: Classification performance for LDA with feature selection/reduction modules with 10 features on unbalanced easy data (SNR = 3.0)

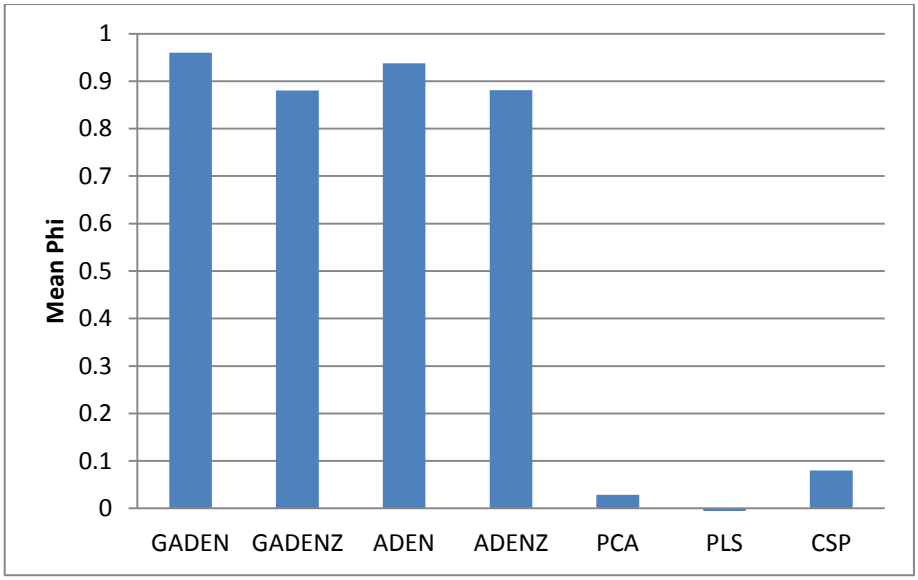

Figure 6.2: Classification performance for LDA with feature selection/reduction modules with 10 features on unbalanced hard data (SNR = 0.3)

For unbalanced datasets, GADEN and ADEN were found to yield the highest overall performance for several metrics. Figs. 6.1 and 6.2 detail a specific case, the results of LDA combined with various feature selection modules. Fig. 6.1 shows the easy (SNR = 3.0) data, where each module functions properly. PLS, CSP, and PCA did not yield the consistent performance of ADEN, ADENZ, GADEN, and GADENZ across each performance metric. System configurations utilizing ADEN, ADENZ, GADEN, and GADENZ were the only methods able to successfully classify the hard dataset (SNR = 0.3) of both balanced and unbalanced data. The max phi of GADEN with 10 features was 0.96. No system configuration was able to correctly classify the balanced or unbalanced very hard datasets (SNR = 0.03). PCA dropped in performance greatly when faced with the hard dataset (SNR = 0.3), in contrast to $ADEN_1$.

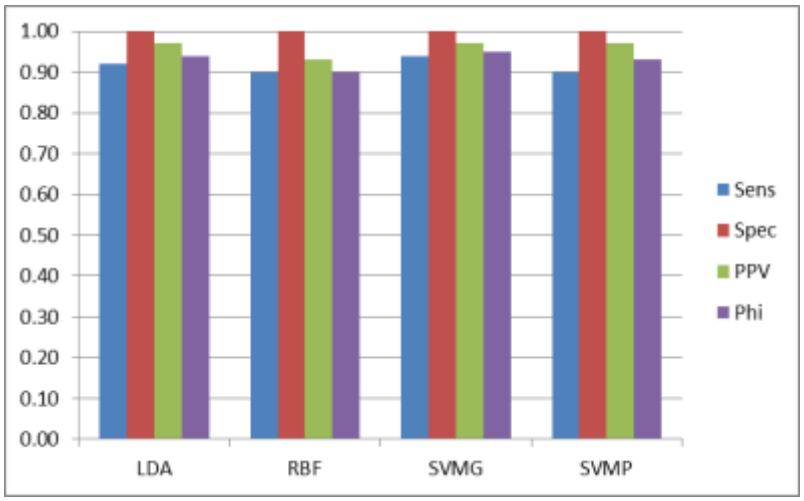

Figure 6.3: Classification performance for 10 ADEN features with pattern recognition modules on unbalanced hard data (SNR = 0.3)

Performance metrics were high across all pattern recognition modules for the unbalanced hard dataset, as detailed in Fig. 6.3. With ADEN, high performance was independent of the pattern recognition module used. However, performance metrics varied greatly across datasets. As expected, a general performance trend in the pattern recognition modules was a drop as the signal grew weaker relative to the background EEG. However, a specific counter-example (SNR = 0.3) is shown in Fig. 6.5. The upswing only occurred with use of ADEN, and was not significant (p = 0.19), A steady downward trend was present in ADENZ even at the hard (SNR = 0.3) data.

Classifier performance metrics, including sensitivity, selectivity, and phi, dropped as the SNR went from 16 to 0.03. Fig. 6.4 presents a specific module configuration, ADEN with LDA on the unbalanced data, to depict the drop at 0.03.

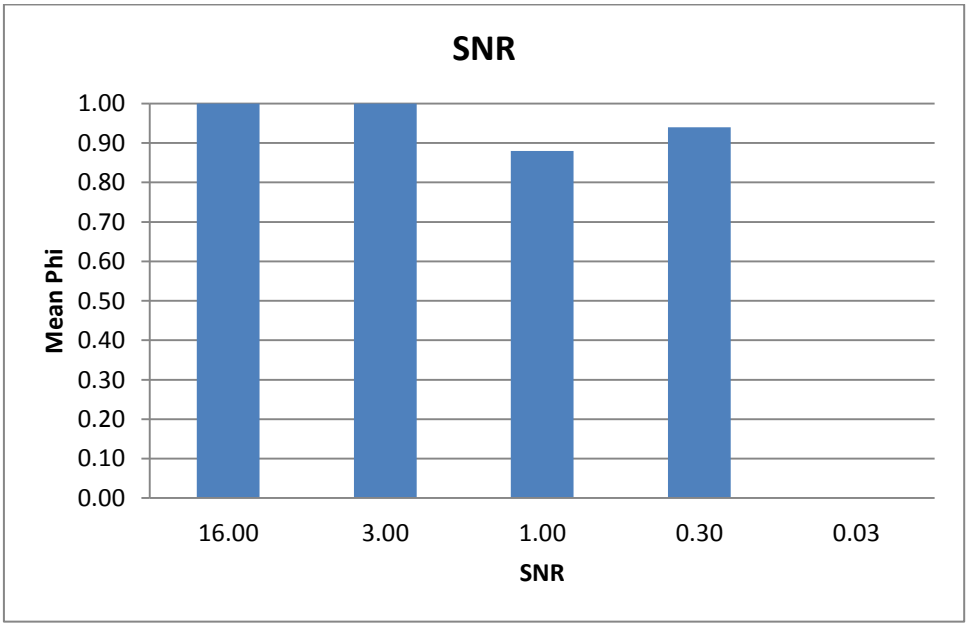

Figure 6.4: Classification performance for $ADEN_{10}$-LDA on unbalanced data (SNR = 16.0 to 0.03)

The upswing witnessed in Fig. 6.4 on the hard (SNR = 0.3) data does not represent the typical case. Instead, the typical case is presented in Fig. 6.5 with a specific module configuration, PCA with LDA on the unbalanced data.

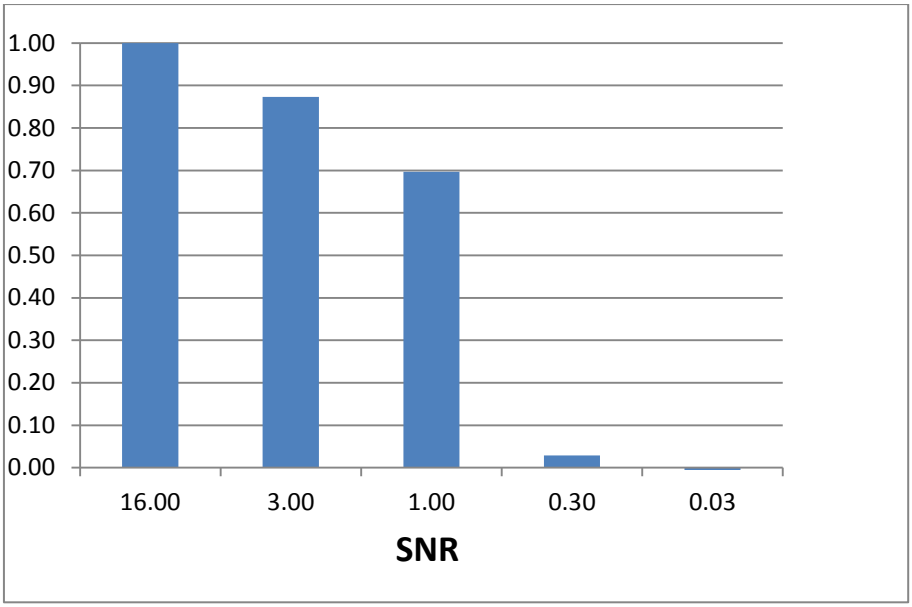

Figure 6.5: Classification performance for PCA10-LDA on unbalanced data (SNR = 16.0 to 0.03)

The drop in performance occurred independently of whether the data was balanced or unbalanced. In addition, only combinations of modules incorporating $ADEN_1$ managed to successfully classify the hard dataset (SNR = 0.3) on the balanced dataset above random guessing. The highest phi for that task came from the Gaussian SVM kernel combined with ADEN, with a phi of 0.95 on the hard balanced data. No module combination was able to correctly identify the majority of events and non-events for the "very hard" unbalanced dataset.

6.3.2 Ensemble Classifier Performance

Ensembles were investigated for their performance relative to single classifiers. However, ensembles did not provide the hoped for improvement in many cases, given the relatively high results of the single classifier system. System configurations consisting of ADEN as a feature reduction method combined with LDA as a pattern recognition method, tested upon the hard data, were compared across ensemble structures.

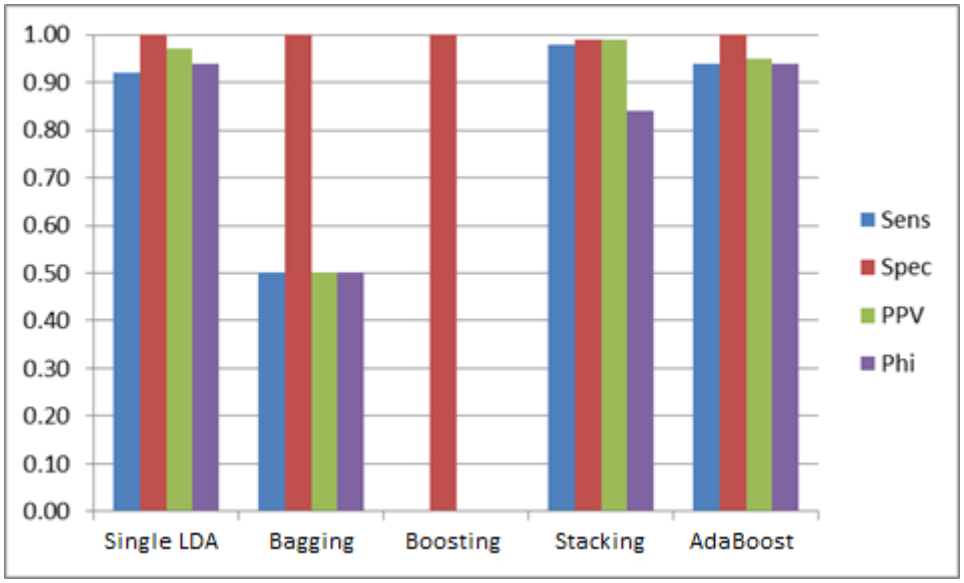

Figure 6.6: Classification performance for $ADEN_{10}$ LDA on unbalanced hard (SNR = 0.3) data

As shown in Fig. 6.6 on the unbalanced hard dataset, performance for each ensemble structure was different. AdaBoost (with three weak learners) again performed consistently well, with stacking behind it in terms of phi value. However, AdaBoost was roughly equivalent in performance to a single classifier system. A high specificity is of no value when combined with a poor selectivity or sensitivity. While phi was lower on stacking than with single classifier cross-validation, stacking had the highest sensitivity of the ensemble systems. The balanced data yielded similar results.

6.4 Discussion

Despite the variety of artificial datasets and system configurations investigated, key trends were noted. Average distance feature selection modules provided higher performance scores than other FS/R algorithms, independent of class balance or pattern recognition module. The presence of an ensemble system did not change this. However, $ADEN_1$, $ADEN_{10}$, or $GADEN_{10}$ could not successfully classify the very hard dataset (SNR = 0.03), whether it was balanced or unbalanced.

While not presented in this chapter, ADEN also performed the highest upon the balanced datasets, with GADEN a close second. However, differences in performance were not as high as in the unbalanced datasets. This may be because averaging a larger number of non-events (such as present in the unbalanced data) better decreases background noise than a smaller subset (such as in the balanced data).

As the event signal decreased in amplitude, it became harder to discern from the background EEG. An upswing in performance was noted with $ADEN_{10}$ in the hard dataset (SNR = 0.3) relative to the medium (SNR = 1.0) dataset, but it was only marginally higher

than performance on the higher SNR datasets. $ADEN_{10}$ had a phi value of 0.94 (0.77-1.00) on the hard dataset (SNR = 0.3), and the medium data had a phi of 0.88 (0.70-1.00). The difference was not significant (p = 0.19). The slight upswing only occurred in $GADEN_{10}$ and $ADEN_{10}$. With $ADENZ_{10}$ and $GADENZ_{10}$, the mean phi for the hard (SNR = 0.3) data was 0.88 (0.54-1.00), while phi for the medium dataset was 0.98 (0.86-1.00).

However, neither $ADEN_1$ nor $ADEN_{10}$ could correctly classify the "very hard" artificial data (SNR = 0.03) in either the balanced or unbalanced case. The amplitude of the event may have dropped to a point of being indistinguishable from the background noise.

Also of note was the clear dominance of average distance feature selection methods over alternatives. The results indicate that performance is strongly dependent upon feature selection rather than type of classifier or class balance. Across pattern recognition modules, LDA, RBF, and SVMs with different kernels yielded similar performance to across the same datasets.

A hypothesis was that ensembles would perform noticeably higher than single classifiers in almost all cases. So far, this has been shown not to be the case, for balanced or unbalanced data. The single classifier and ensemble could correctly identify few of the simulated events in the very hard (SNR = 0.03) unbalanced data. While statistical significance tests were not performed, many of the results were only slightly or not higher than single classifiers. Of the varieties of ensembles, stacking and AdaBoost provided consistently high performance metrics.

On the artificial data, ADEN with any single classifier or ensemble yielded the highest consistent performance across datasets, as detailed in the Appendix and shown in Fig. 6.2. The corresponding spectral features for 15 Hz increased relative to other spectral bands, turning the task into a thresholding problem. Given the artificial event increases upon a specific spectral band, average distance feature selection may be better able to determine where to set such a threshold.

PCA dropped in performance on the hard data (SNR = 0.3), while ADEN did not. This suggests that generation of meta-features, as opposed to selecting a subset of existing features, may lose microsleep-relevant information. While PCA selects meta-features that may be uncorrelated to the target event, ADEN can select features based on *a priori* knowledge of class differences. It is likely that multiple ADEN features would contain redundant information, but this may be advantageous. It may combine signals corresponding to the same event across multiple channels, increasing the probability of successful detection. The principal components found by PCA are combinations of multiple features, many of

which could be noise. Average distance feature selection methods may prove more suitable for the microsleep detection task.

6.5 Summary

Classifier performance on the simulated EEG feature sets resulted in high mean phi results of 0.95 for ADEN and 0.96 for GADEN on SNR = 0.3. Despite this success, microsleep identification is likely to be substantially more difficult than the 0.3 feature set. As PCA was unable to match ADEN's performance on the 0.3 feature set, supervised feature selection requires a more thorough investigation to improve the performance of microsleep detection. The 15-Hz simulated event was substantially more consistent than EEG, so all system configurations investigated had to be re-evaluated before any could be eliminated.

CHAPTER 7. REPLICATION OF PRIOR BENCHMARKS

7.1 Introduction

After validating the system on artificial data, replication of prior performance benchmarks was an essential precondition of further analysis. The highest performance in microsleep detection reported was a mean phi value of 0.39 using LDA with a stacking ensemble (Peiris et al., 2011). This was the equivalent of the SABIL feature set (N = 8), which as mentioned in Section 5.3, which had undergone ICA-based removal of eye movement and artefact pruning, the latter removing entire segments of data from training and testing. Therefore, before varying the preprocessing steps or system configuration, system replication was a prerequisite.

7.2 Methods

The first step of the process was the reconstruction of the same feature set and system used in the prior study. The SABIL feature set was used with an LDA-based stacking ensemble and PCA was employed for feature reduction, as described by Peiris et al. (2011). The system used seven subjects to train the ensemble, and one to test it. The process was repeated eight times, with the mean phi value being the primary performance metric. Peiris et al. (2011) mentioned that 50s PCs corresponded to the highest performance. To explore more thoroughly, the number of principal components was varied. It was inferred that the system's highest values would be close to the benchmark. In order to provide additional validation, the use of a single LDA classifier with PCA was also examined and compared to prior work (Peiris et al., 2011).

7.3 Results

The stacking ensemble values were slightly higher than the reported value of 0.39, while the single classifier values were slightly lower than 0.31 (Peiris et al., 2011). When replicating the prior work with the stacking ensemble, the highest mean phi value was 0.40 (0.13-0.66) with 150 PCs.

The highest value for a single LDA classifier was a mean phi of 0.30 (0.03-0.67) with 160 PCs. For the single classifier, the highest mean value was at the end of a plateau of values beginning at 40 PCs at a mean phi of 0.28 (0.07-0.58).

7.4 Discussion

The replication of prior work using Study A results was used as a baseline to presage variant training and testing scenarios for Studies A and C. With the SABIL feature set, the stacking ensemble and single LDA classifier generated the same results as reported earlier (Peiris et al., 2011). The optimal number of PCs is less than 200 in each case, although the number may be different for other features. The mean phi value of 0.23 (0.04-0.49) with 10 PCs with a single LDA classifier was not as high as the stacking ensemble's value of 0.33 (0.11-0.52) for 10 PCs, indicating that the stacking ensemble can improve performance if a single linear classifier can achieve some success.

The SABIL feature set was only investigated in a single system configuration, i.e. a combination of PCA feature reduction with an LDA classifier. The only factors changed were the numbers of PCs and use of a single LDA classifier or stacking ensemble. Variations in preprocessing of the Study A data and different system configurations were also fully examined in later chapters.

7.5 Summary

The replication of prior work was essential to build a basis for further work. With the SABIL features, the stacking ensemble with PCA yielded a maximum mean phi correlation of 0.40 (0.13-0.66) with 150 PCs. The highest value for a single LDA classifier was a mean phi of 0.30 (0.03-0.67) with 160 PCs. After successfully replicating prior work, investigation of other system configurations and preprocessing methods with Study A data was necessary.

CHAPTER 8. PRELIMINARY ANALYSIS OF FEATURE SET PREPROCESSING AND TRAINING SCENARIOS

8.1 Introduction

After the replication of Study A's prior benchmark, changing the preprocessing steps resulted in drastically different feature sets. The SABIL feature set included ICA-based eye blink removal and artefact pruning prior to feature extraction, and the results reported used the same feature set for LOOCV. As a result, the feature set was considered to be "cleaner" than EEG data without ICA or artefact pruning. Davidson et al. (2007) reported that ICA did not have a major effect on results, and there was no pruning in his evaluation. The effect of not artefact pruning was not reported by Peiris et al. (2011), nor was the concept of training on different variants of the same dataset.

As each dataset took substantial time to comprehensively examine, a method of reducing the total variants to examine was devised. Factors such as ICA preprocessing, artefact pruning, bipolar conversion, and feature extraction method were varied to create variants for each. If change in a variable did not improve performance on a single LDA classifier-based LOOCV case, the corresponding variant was removed. Only a select few variants of Study A were retained, and the preprocessing findings were applied to Study C.

In addition to Study A, Study C (N = 10) required preliminary exploration due to the lack of a prior performance benchmark. A challenge in directly applying the feature extraction methodology from Study C was the differing number of channels between subjects with only 19 common channels for all subjects. The variant methods of feature extraction and interpolation methods were applied to Study C to determine the optimal feature set to examine in future work.

A concept unexplored in prior work was the possibility of having different feature sets for training and testing. For example, a classifier could be trained on a "cleaner" feature set having undergone ICA and artefact pruning and tested on a feature set without it. The possibility of training on a balanced version of the feature set, or an alternative gold standard based on definite microsleeps, rather than lapses had also not been previously explored. These possibilities were explored with the original feature set and system, so as to eliminate them early if they would prove unhelpful even in a "best case scenario."

8.2 Methods

8.2.1 Variant Study A Scenarios

8.2.1.1 Changes in Feature Extraction

In order to directly compare the effects of varying feature extraction methods, the performance with the SABIS features was compared with the SABIL features. The SABIS feature set was used alongside an LDA-based stacking ensemble with PCA used for feature reduction, as used by Peiris et al. (2011). The system used LOOCV with the mean phi value being the primary performance metric.

As with the replication of earlier work, the number of PCs was varied to find those corresponding to the highest performance (Peiris et al., 2011). It was suspected that the system's highest values would be close to the benchmark. In order to provide additional validation, the use of a single LDA classifier with PCA was also examined and compared to prior work with the SABIL features (Peiris et al., 2011). It was thought that the SABIL features would be more robust due to clearly showing non-linear changes than the SABIS features.

8.2.1.2 Changes in Preprocessing

The omission of preprocessing steps was also compared. Two feature sets without ICA and artefact pruning were generated, one bipolar and the other referential. Both feature extraction methods were compared against each other for the bipolar (SABUS) and referential (SARUS) sets, as it was possible that one feature extraction method might be affected by the exclusion of ICA and artefact pruning.

The initial SARUS and SABUS feature sets were generated using the same feature extraction method as the SABIS features, average spectral power. Variants of the SARUS and SABUS feature sets using the same log power feature extraction method as the SABIL features and additionally compared. The SARUL and SABUL features were compared with the SABIS and SABIL features. Other system configurations, including FS/R methods like ADEN, ADENZ, and PLS were investigated alongside PCA with a single LDA classifier for this and the following phases.

8.2.1.3 Changes in Training Data Balance

Variant scenarios involving different microsleep training scenarios were then examined. A training scenario involving training on a balanced feature set and testing on an unbalanced feature set was examined using the SABIL feature set with the same system

configuration used previously. Further training variations in balance were studied later, but the initial test was performed to see if the concept performed no worse than standard LOOCV.

8.2.1.4 Changes in Testing Data

The concept of testing on a different feature set compared to a training feature set was also examined. A "cleaner" feature set in terms of ICA and artefact pruning used for training could better instruct a classifier than a feature set without ICA and artefact pruning. In order to test the concept, the SABIL feature set was used to train a classifier, while the SARUS and SABUS features were tested. Even if no performance improvements occurred, the findings could demonstrate that a microsleep detector could function upon features less preprocessed than it was exposed to during training. If unsuccessful, the alternative feature set training methodology would be dropped.

8.2.1.5 Changes in Gold Standard

To conclude the initial Study A investigations and tests, the performances of four different gold standards were compared with the SABIL and SABIS feature sets: lapses in responsiveness, flat spots, video microsleeps, and definite microsleeps. The definite microsleeps gold standard represented a reduction in the total number of events relative to the lapses in responsiveness gold standard, but might offer increased performance. The results were used to determine which to use in future tests.

8.2.2 Study C Preprocessing Comparison

For Study C, the referential EEG feature set was examined using the exact feature extraction method used in Study A. Spherical and inverse distance interpolation were used via EEGLAB (Delorme, 2004) and compared with the original "null channel" EEG feature set. A comparison was made between taking linear spectral features and the log spectral features for performance, with and without interpolation. A single LDA classifier with four FS/R modules (ADEN, ADENZ, PCA, and PLS) was used with LOOCV for the Study C tests to provide a baseline for future tests involving classifier ensembles. If no variant Study C feature set achieved a mean phi value on a simple classifier above random guessing, then further analysis into Study C would be required.

8.3 Results

8.3.1 Variant Study A Scenarios

8.3.1.1 Changes in Feature Extraction

The SABIS features performed lower on both the stacking ensemble and with the single LDA classifier. With the stacking ensemble, the SABIS features performance peaked at 100 PCs at a phi value of 0.36 (0.14-0.63). As show in Fig. 8.1, the highest mean phi values from the single LDA classifier on the SABIS feature set, 0.27 (0.00-0.51), was lower than the SABIL feature set's max phi of 0.33 with 10 ADENZ features.

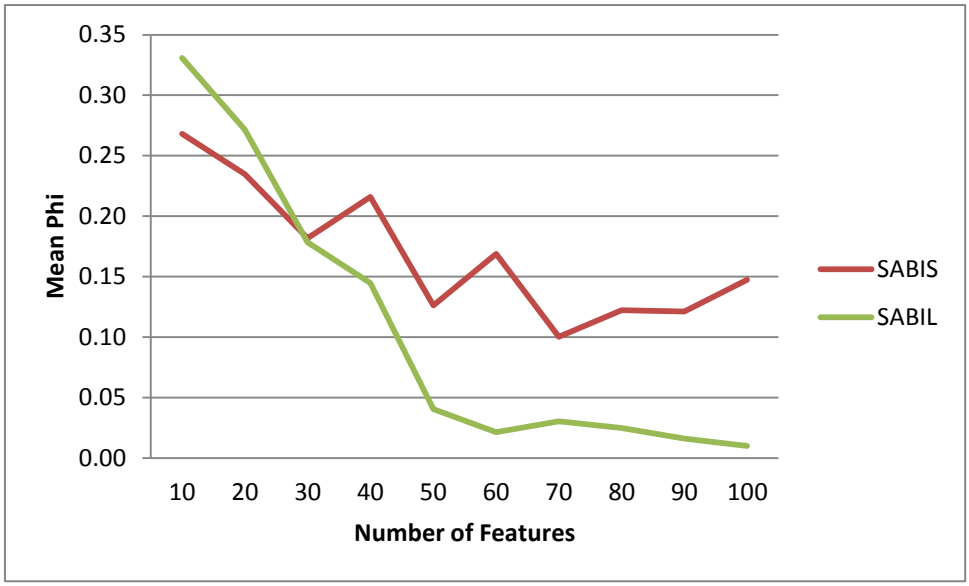

Figure 8.2: Comparison of SARUS and SABUS features with ADENZ and single LDA LOOCV

For other values, SABIL features continually outperformed SABIS in terms of mean phi, sensitivity, and selectivity values. Fig. 8.2 shows how the SABIL features outperformed other feature sets.

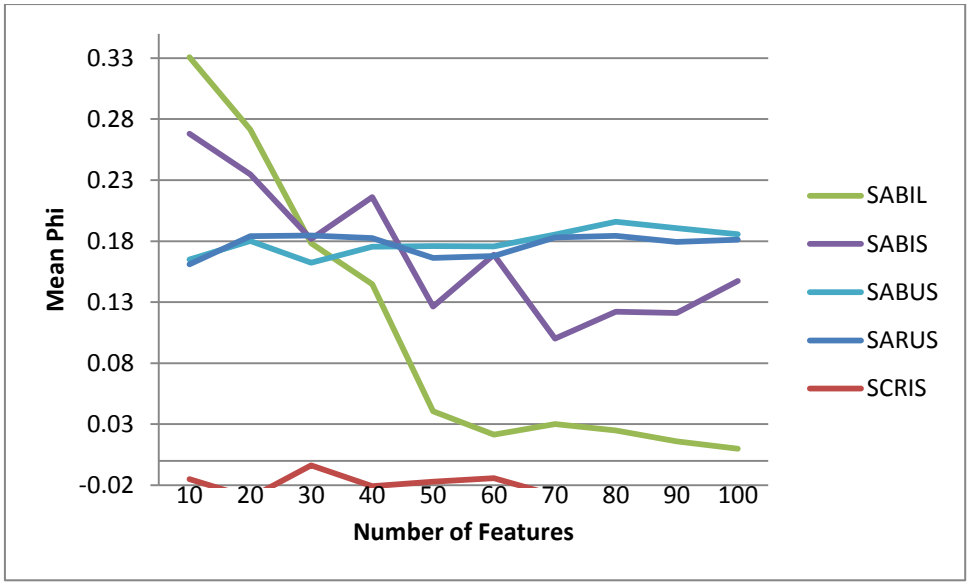

Figure 8.2: Comparative performance of ADENZ with a single LDA classifier on unbalanced major feature sets

Both SABIL and SABIS were the highest performing feature sets in terms of phi correlation.

8.3.1.2 Changes in Preprocessing

Both variants of the SARUS and SABUS feature sets performed lower than the SABIS and SABIL features in most cases. The highest mean phi value for the SABIL features corresponded to 10 ADENZ features with 0.33 (0.12-0.52), but the highest value of the SABUS features was a mean phi of 0.27 (0.02-0.56) with 80 ADEN features. As shown in Fig. 8.3, the highest mean phi value of the SARUS features was 0.26 (0.05-0.43) with 30 ADEN features.

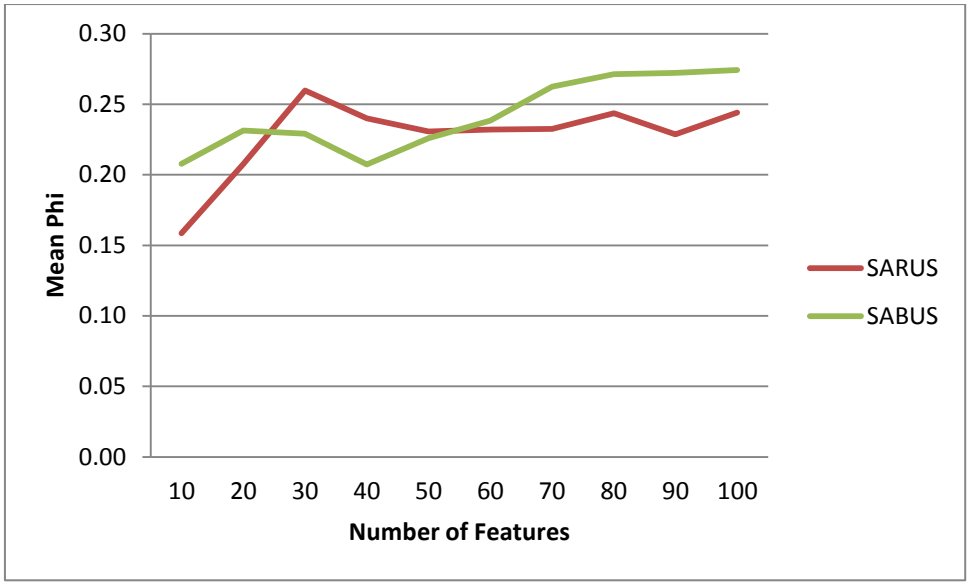

Figure 8.3: Comparison of SARUS and SABUS features with ADEN and single LDA LOOCV

When the original SABUS and SARUS feature sets were compared with variants using log of the power spectrum feature extraction, no statistically significant changes occurred. The highest mean phi value for either SARUL or SABUL was with SABUL features achieving a mean phi of 0.35 (0.12-0.51) with 150 PCs on the stacking ensemble. The highest mean phi value was 0.29 (0.06-0.54) with 70 PCs with the SARUL features with a single LDA classifier. The highest mean phi value on a single classifier for the log power variant of the SABUS feature set, SABUL, was 0.28 (0.05-0.56) with 20 ADEN features.

8.3.1.3 Changes in Balanced Training Data

When training on the balanced SABIL features and testing on unbalanced features, the highest mean phi value of 0.30 (0.03-0.67) corresponded to the single LDA classifier with 160 PCs. The results were equivalent to the standard case of single classifier LDA trained on unbalanced data.

8.3.1.4 Changes in Testing Data

Training the classifier on the SABIL dataset and testing on the SARUS and SABUS datasets did not improve mean phi values above the values reported on standard LOOCV. The highest mean phi value for both SARUS and SABUS datasets was 0.09 (0.02-0.31) with 70 PCs with SARUS. The highest mean phi value for SABUS was 0.04 (-0.11-0.45) with 90 PCs. In all cases, the results were lower than cases trained and tested on unbalanced features of the same type.

8.3.1.5 Changes in Gold Standard

Changing the gold standard did not improve results. As the gold standard was changed from lapses to video microsleeps, the mean phi value with 10 PCs on a single classifier with the SABIS features dropped to 0.24 (0.02-0.54). When the gold standard changed to flat spots, the mean phi value became 0.20 (0.01-0.49). When using definite microsleeps only, the phi value was 0.21 (0.03-0.50). The most dramatic drop occurred with 10 ADENZ features on the SABIS data. A mean phi value of 0.27 (0.00-0.51) on the lapse gold standard dropped to 0.18 (0.02-0.36) on the definite microsleep gold standard. Results were similar with the SABIL dataset, as each definite microsleep case was lower than the corresponding lapse gold standard case.

8.3.2 Study C Preprocessing Comparison

For Study C, mean phi performance values were low on both the stacking ensemble and single LDA classifier on the SCRIS spectral band features. With the single LDA classifier, the highest mean phi was 0.10 (0.00-0.32) with 10 ADEN features. With the stacking ensemble, the highest mean phi value was 0.10 (-0.13-0.12) with 10 PCs.

Interpolation techniques applied to the referential EEG did not improve classification results. The highest mean phi value for spherical interpolation features, SCRILSP, was 0.07 (-0.02-0.19) with 50 ADENZ features with the single LDA classifier. The highest mean phi value for inverse distance interpolation features, SCRILID, was 0.06 (-0.07-0.30) with 30 ADENZ features with the single LDA classifier. For the SCRISSP features, the highest mean phi was 0.03 (-0.18-0. 24) with 10 ADEN features. For the SCRISID features, the highest mean phi was 0.04 (-0.03-0.10) with 10 PLS features.

By using the log power SCRIL features, the highest mean phi value was 0.01 (-0.07-0.07) with 10 ADENZ features and a single LDA classifier. The low results led to the dropping of the interpolated SCRISSP, SCRISID, SCRILSP, and SCRILID features and the SCRIL features for Study C, due to the decision to keep additional processing minimal.

8.4 Discussion

The replication of prior work with Study A was used to presage variant training and testing scenarios for Studies A and C. With the SABIL feature set, the stacking ensemble and single LDA classifier generated the same results as reported earlier (Peiris et al., 2011). The optimal number of PCs is less than 200 in each case, although the number may be different for other features. A mean phi value of 0.23 (0.04-0.49) with 10 PCs with a single LDA classifier was not as high as the stacking ensemble's value of 0.33 (0.11-0.52) for 10 PCs,

indicating that the stacking ensemble can improve performance if a single linear classifier can achieve some success.

Changes in preprocessing drastically affected results. The SABUS and SARUS features performed notably lower in the key performance metrics on the same system configurations. While prior work reported a mean phi value of 0.38 without ICA and without artefact pruning, this was achieved utilizing a specialized neural net instead of a single classifier or rudimentary ensemble (Davidson et al., 2007). However, changing the feature extraction method did not improve as much had been expected from performance on artificial event data, and mean phi values were not enough to surpass even the SABIS feature set's performance values. No significant differences were found between the maximum mean phi performances of SABIL, SABIS, SARUL, and even the "raw" SARUS and SABUS feature sets ($p > 0.15$ in all cases). As such, the standard spectral power SARUS and SABUS feature sets were retained. As shown in Table 8.1, the highest mean phi values did not surpass the SABIL features with PCA and the stacking ensemble.

Table 8.1: Maximum mean phi values and system configurations

Set	Mean Phi	Min Phi	Max Phi	FS/R	Classifier	Structure
SABIL	0.40	0.13	0.66	PCA150	LDA	Stacking
SABIS	0.36	0.14	0.63	PCA100	LDA	Stacking
SABUL	0.35	0.12	0.51	PCA150	LDA	Stacking
SABUS	0.33	0.12	0.52	ADENZ10	LDA	Single
SARUS	0.27	0.02	0.56	ADEN80	LDA	Single
SARUL	0.29	0.06	0.54	PCA70	LDA	Single
SCRIS	0.10	0.00	0.32	ADEN10	LDA	Single

Variations on the basic training and testing regime proved inconclusive with regards to balancing with Study A. The SABIS features performed notably lower than the SABIL features, indicating that taking the log of the power spectrum is a more effective feature extraction method. Further comparison between the SABIS and SABIL features was undertaken to determine if the trend continued.

Further investigation of the SABIS feature set was necessary to determine if its performance could be brought up to match the SABIL feature set, as well as how the SABIL feature set performed on other system configurations. While the SABUL features were behind the SABIS and SABIL features, it was unlikely that the unprocessed features would consistently outperform the SABIL and SABIS features.

The SARUS and SABUS feature sets were retained in order to directly compare the referential and bipolar features with a minimum of preprocessing. As the SCRIS features were referential and used spectral features, the Study A feature sets corresponding to changes no more than two variables (e.g., referential or bipolar, ICA preprocessed or unprocessed, and spectral or log spectral features) away were retained. As such, the log power SABUL and SARUL feature sets were dropped. The SARUS and SABUS features were retained for contrast with the SABIL and SABIS features.

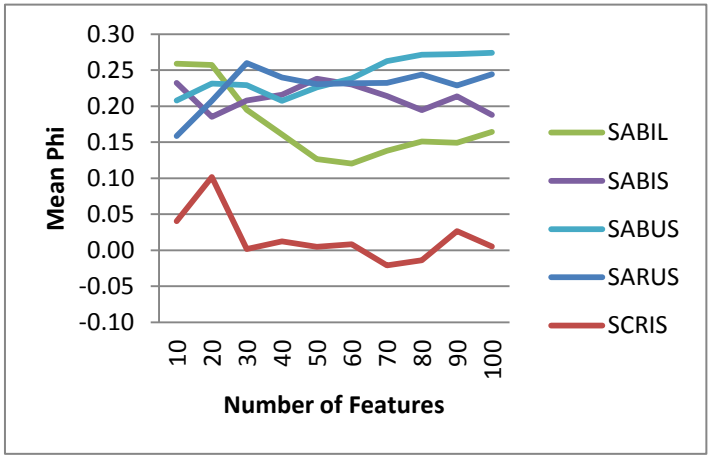

Figure 8.3: Comparative performance of ADEN with a single LDA classifier on major feature sets

Balancing the feature set resulted in no improvements to the classification results. Removal of the bias in training did not result in the performance increases. However, the results were no worse than the control case of training on unbalanced features, or features with one class as a small minority of total observations. The ambiguous results required further and more thorough analysis on balanced features, features where the ratio between classes is unity. It was thought that other system configurations could potentially benefit from balanced training features more than the limited scenarios evaluated.

Training a classifier on a different feature set than it was tested on resulted in poor performance. Differences between the SABIL, SABIS, SARUS, and SABUS features might have been too great to allow for successful classification. However, the concept of developing a specific variant of a feature set, for training purposes only, was one that warranted further investigation.

The use of the lapse gold standard resulted in higher mean phi values than the definite microsleep gold standard. The use of the definite microsleeps gold standard had fewer events per individual than the lapse criterion. As such, the training imbalance might have been further exacerbated. While further analysis might improve results, the lapse gold standard

was used in future work so direct comparisons with previous work could be made (Peiris et al., 2011), as well as keeping the number of events constant.

The performance results from Study C corresponded to random guessing in each case. While the gold standard of Study C was superior to Study A, the EEG was thought to be insufficient in quality. The causes were thought to be that Study C had underwent several steps involving removal of data on already noisy EEG channels. The steps potentially resulted in less useful information on the channels that remained, and potentially that the specific system configuration could be sub-optimal for the feature set.

Unlike Study A, the gold standard used for Study C did not include "lapses," but only "definite microsleeps" and "sleeps" (> 15 s) as events. It was considered that if a system configuration could successful classify the basic SCRIS feature set, it would be investigated more closely. Due to the limitations of the primary Study C EEG dataset used, the SCRIS features were considered the most challenging classification task.

8.5 Summary

The replication of prior work was essential to build a basis for further work, but the variations in feature sets had to be reduced to a smaller, more promising number. As expected, in contrast to SABIL features, the SABIS features resulted in lower mean phi performance values on both the stacking ensemble, 0.36 (0.14-0.63), and single LDA classifier, 0.27 (0.00-0.51). Not using ICA and artefact pruning, as with the SABUS and SARUS feature sets, did not result in significant performance decreases ($p > 0.15$). The use of training on balanced features yielded inconclusive results. Training and testing on different feature sets resulted in poor performance. Choosing a gold standard to only definite microsleeps rather than lapses resulted in decreased performance. With the Study C features, SCRIS, the highest mean phi was 0.10 (0.00-0.32) with 10 ADEN features with a single classifier. All Study C variant features were ultimately derived from the same EEG, only SCRIS was retained. Despite the lack of performance increases, the feature sets were reduced to SABIL, SABIS, SARUS, and SABUS for Study A and SCRIS for Study C. Due to the many variables, further analysis was necessary for more compelling results on Studies A and C.

CHAPTER 9. SELECTION OF OPTIMAL SYSTEM CONFIGURATIONS THROUGH EVALUATION FOR MICROSLEEP DETECTION

9.1 Introduction

With the fundamental modules of ICTOMI implemented and validated, analysis of expert-rated EEG datasets commenced. Due to the variety of permutations and combinations of modules, the least promising would be eliminated to leave only the most promising system configurations. To validate the modules lacking prior benchmarks in the literature, they would be compared against other implementations.

While benchmarks for microsleep detection using single classifiers and stacking existed (Peiris et al., 2011), benchmarks for AdaBoost, bagging, and boosting did not exist. Another toolbox, the University of Waikato's WEKA toolbox (Hall, 2009), was compared directly with ICTOMI. In particular, the classifier ensemble systems were ones prioritized for ensuring proper implementation.

After the validation of ensemble modules, reduction of system configurations for further research commenced. As previously detailed (Chapter 4), ICTOMI consisted of different modules for feature extraction, FS/R, pattern recognition, and classifier structure. Due to a total of 49 possible system configurations (excluding feature extraction), the number had to be reduced to an optimal number of the most promising systems suitable for analysis. By giving different configurations access to the same feature sets, the results could be directly compared.

9.2 Methods

Additional validation was required for the other ensemble modules before system configurations could be compared. In order to compare system configurations against each other, different categories of systems were examined in isolation: feature extraction, FS/R, pattern recognition, and classifier structures. The systems with the highest mean phi correlations would be kept, and the ones that with the lowest would be removed. For each phase, the two most promising feature sets were selected: the SABIL and SABIS features. Each provided the benefit of comparing a different method of feature extraction. Before comparing systems, the ensembles had to be validated.

9.2.1 Validation of Ensembles with WEKA

Due to possible differences in the implementation with the prior system (Peiris et al., 2011), the WEKA toolset from the University of Waikato was used to provide additional validation of the results. The WEKA toolset has been used in many areas of research and offered a peer-reviewed, accessible software package permitting a large battery of tests upon feature sets (Hall, 2009). Any differences in performance between common classifier ensembles present in ICTOMI and WEKA would be compared to provide additional validation of prior results.

The analysis performed was the SABIS feature set was initiated with 10 PCs, with the standard "lapses of responsiveness" gold standard listing both video microsleeps and tracking "flat spots" as events. Standard LOOCV was used, with the mean phi results from stacking, bagging, and AdaBoost compared against their ICTOMI implementations. Equivalent results between ICTOMI and WEKA would imply the ICTOMI implementations were well-suited for the classification task.

9.2.2 Module Performance Comparison

The two best performing feature sets in Chapter 8, SABIL and SABIS, were used to compare FS/R, pattern recognition, and classifier structure modules sequentially.

9.2.2.1 Comparison of Feature Selection/Reduction Modules

For the FS/R, modules for PCA, PLS, CSP, GA, and the ADEN variants were tested with a single LDA classifier. Due to the high memory requirements to run GA, three generations of 100 offspring were examined. After the completion of a full simulation, the number of "genes" was incrementally increased from four to 13. After the completion of a full simulation, the number of "genes" was incrementally increased from four to 13. GADEN and GADENZ, by contrast, were limited to a pool of the top 30 ADENs and 150 cumulative offspring. A subset of 10 features or meta-features was used in all cases.

9.2.2.2 Comparison of Pattern Recognition Modules

For the pattern recognition case, modules for LDA, RBF, SVMG, and SVMP were used. If poor performance was limited to a single classification algorithm, independent of FS/R, it would be dropped.

9.2.2.3 Comparison of Classifier Structure Modules

For the classifier structure case, modules for a single LDA classifier, bagging, boosting, stacking, and AdaBoost were compared. LDA was the component classifier for

each of the ensembles. It was assumed that stacking and AdaBoost would perform the highest out of the ensembles, based upon prior work (Peiris et al., 2011; Freund and Schapire, 1997).

9.3 Results

WEKA was compared directly with ICTOMI before comparative analysis of system configurations was undertaken.

9.3.1 Validation of Ensembles with WEKA

The highest mean phi value achieved using definite microsleeps only was 0.19 with 10 PCs on the single LDA classifier.

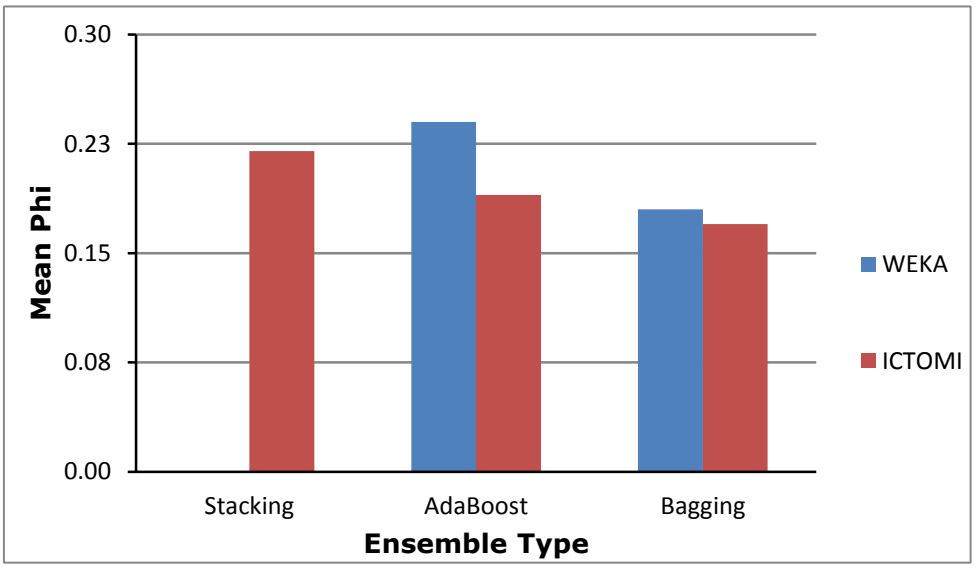

Figure 9.1: All WEKA results with 10 PCs on ensemble systems with SABIS features

The application of WEKA did not result in higher mean phi values on the SABIS feature set. The mean phi value for stacking was 0.00 (with an individual range from -0.01 to 0.02). AdaBoost and bagging achieved higher mean phi values at 0.24 (0.00-0.58) and 0.18 (0.03-0.35), respectively. When compared with the ICTOMI results, WEKA performed slightly higher on AdaBoost and bagging. ICTOMI had a higher mean phi value on stacking.

9.3.2 Module Performance Comparison

After ensemble validation with WEKA, module-based research commenced.

9.3.2.1 Comparison of Feature Selection/Reduction Modules

A single LDA classifier was first tested with the unbalanced data, so as to replicate prior work (Davidson et al., 2007; Peiris et al., 2011). Feature reduction/selection techniques are presented in Fig. 9.2.

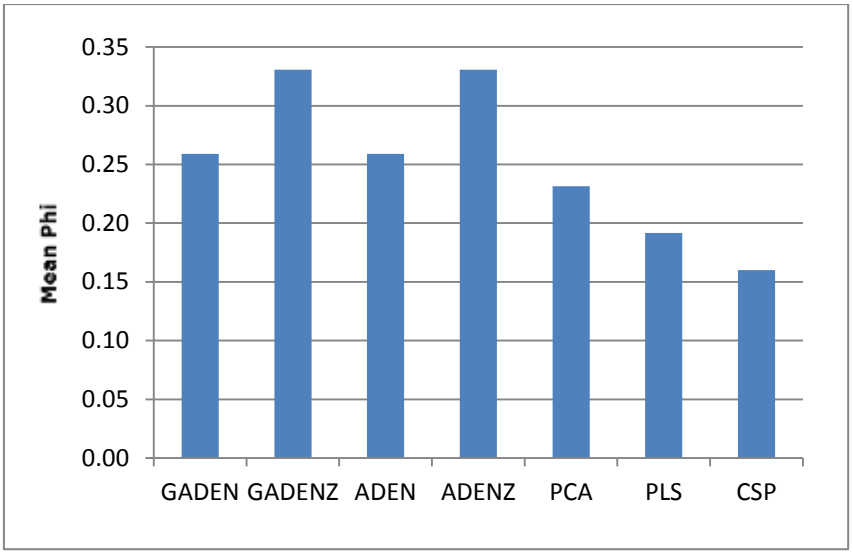

Figure 9.2: Comparison of 10 features on FS/R modules with single LDA on unbalanced SABIL features

With the SABIS features, PCA, ADEN, and ADENZ achieved the highest results with a mean phi value of 0.27. GADEN$_{10}$, limited to the top 30 ADENs and 150 offspring, achieved a mean phi of 0.26. PLS had the lowest mean phi correlation at 0.18. On the SABIL feature set, the highest mean phi value corresponded to GADENZ$_{10}$ and ADENZ$_{10}$ with 0.33, and the lowest was 0.15 with CSP.

As shown in Fig. 9.3, the number of "genes," or features retained for GA, was increased over time.

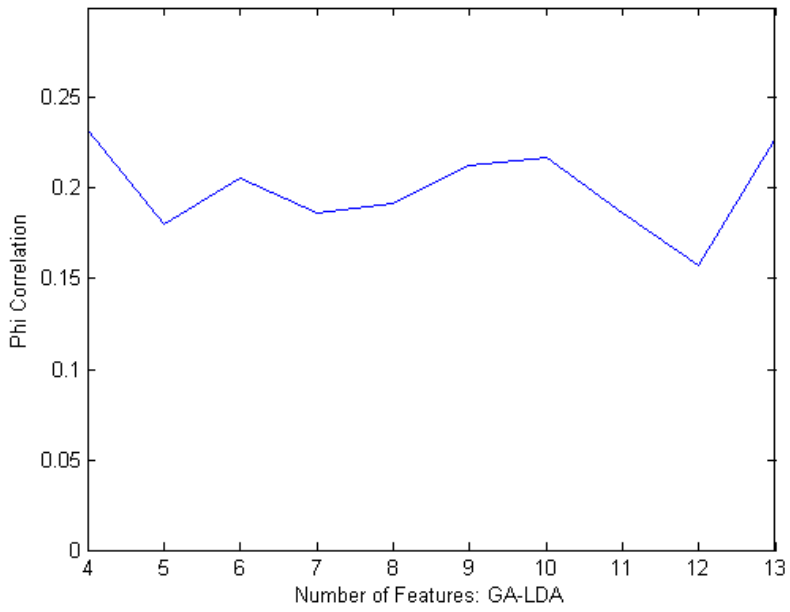

Figure 9.3: GA performance on SABIS features over increased number of features retained

Despite increasing the number of features, performance was erratic. The highest mean phi values, 0.23, occurred with both four and 13 features. Due to performing lower than simple PCA, investigation into GA was not pursued further.

9.3.2.2 Comparison of Pattern Recognition Modules

Single classifiers, such as an RBF neural network and two SVM kernels, were used for direct comparison with LDA.

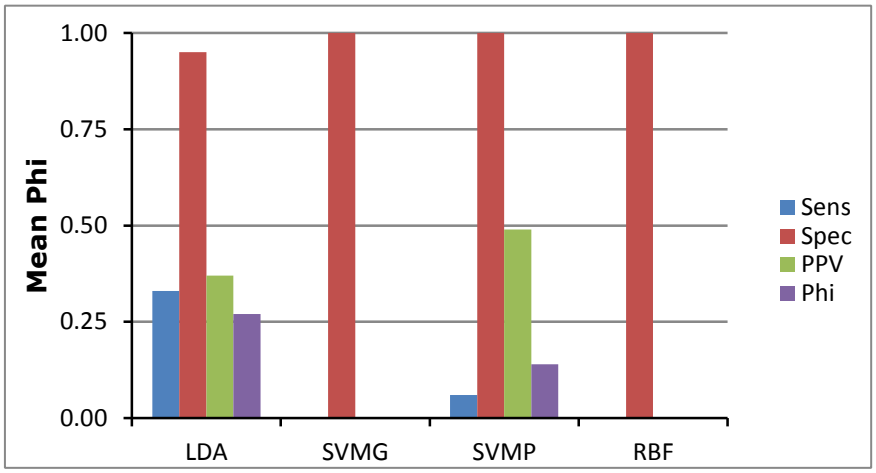

Figure 9.4: Comparison of $ADEN_{10}$ on pattern recognition modules with unbalanced SABIS features

As shown in Fig. 9.4, LDA outperformed other classifiers, with only SVMP also able to partially classify events. The poor performances of SVMG and the radial basis function neural net were unexpected, but the similar operating algorithm of both may explain similar results.

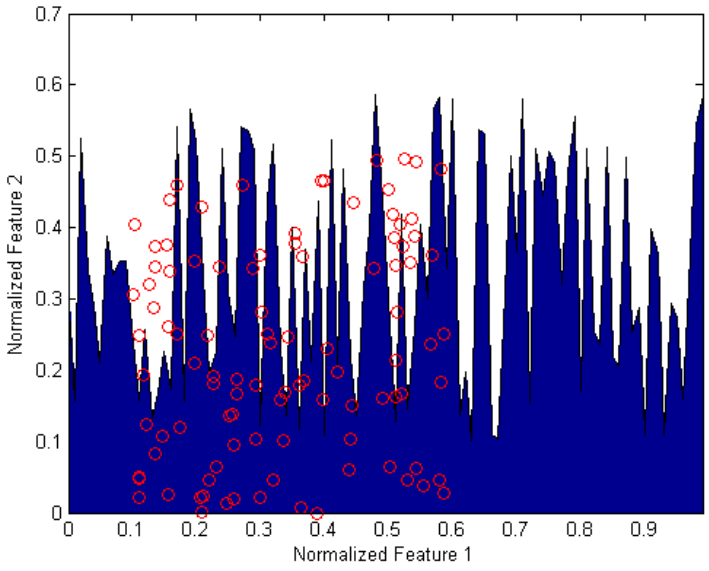

Figure 9.4: Demonstration of SVMG classifier overfitting, with test data as red circles

Fig. 9.5 shows SVMG overfitting on a simplified feature space, with red circles representing testing data. Similar overfitting occurred with RBF and SVMP. Results were similar for the SABIL feature set, with LDA performing the highest at 0.26 on $ADEN_{10}$.

9.3.2.3 Comparison of Classifier Structure Modules

Ensemble classifiers were used to replicate, and potentially improve upon, prior work (Peiris et al., 2011). Fig. 9.6 details $ADEN_{10}$ with a single LDA classifier compared with stacking, bagging, and AdaBoost with 30 weak learners.

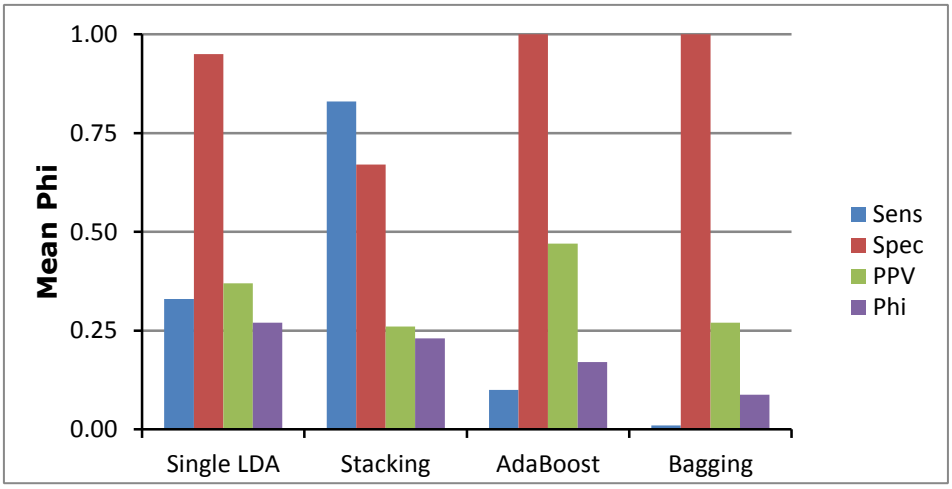

Figure 9.5: Ensemble results of $ADEN_{10}$-LDA on unbalanced SABIS features

Ensembles offered no advantages over a single LDA classifier in the case of 10 ADEN features on the SABIS feature set.

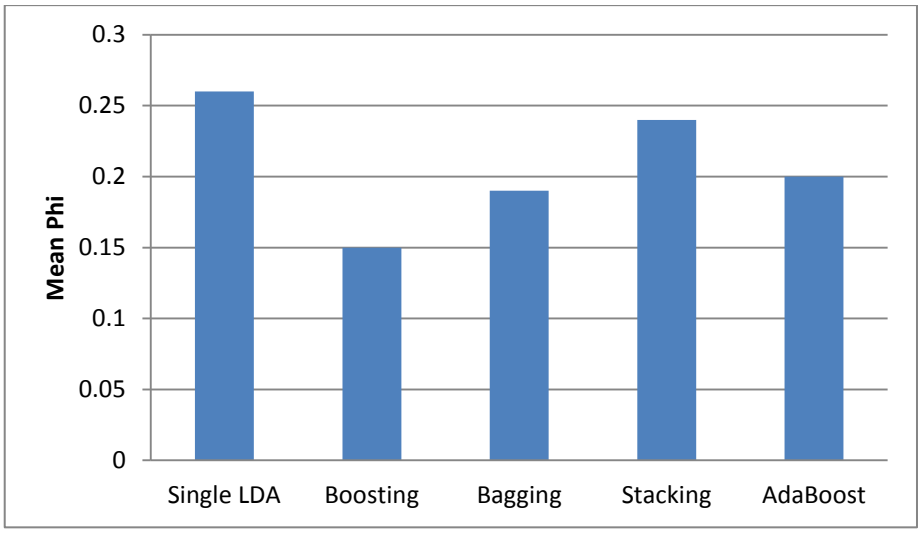

Figure 9.6: Ensemble results of $ADEN_{10}$-LDA on unbalanced SABIL features

For the SABIL features, the unbalanced data achieved a mean phi value of 0.40 with a PCA stacking ensemble, while the maximum single LDA classifier mean phi value was 0.33

with ADENZ$_{10}$. The lowest value was 0.15 with ADEN$_{10}$ on the boosting ensemble. After comparing the results, specific systems were removed.

9.4 Discussion

In addition to the validation of ICTOMI, the results provided a preliminary indication of which approaches could be eliminated.

9.4.1.1 Validation of Ensembles with WEKA

The application of another software toolset did not improve performance on the SABIS features beyond that of ICTOMI. The poor performance of the WEKA stacking module implied issues at dealing with complex feature sets due to its implementation, as it had been validated earlier using artificial event data. Extensive documentation from the WEKA toolset was consulted to ensure it was being used properly, and the results from AdaBoost and bagging largely demonstrate successful classification results.

9.4.1.2 Comparison of Feature Selection/Reduction Modules

By comparing different FS/R modules, PCA and the ADEN variants consistently had the highest phi correlations. CSP's poor performance was considered sufficient for its removal. While PCA and the ADEN variants were largely close, PCA had to be retained due to being used in the prior baseline (Peiris et al., 2011). Notably, supervised FS/R techniques did not seem to result in greater classification accuracy than unsupervised PCA. A potential issue with PLS was that increasing the number of features and model order does not result in performance gains. Despite this, PLS was retained due to being a promising approach to supervised learning relevant to EEG research (Chen, 2013; Hutapea, 2014).

The limitations of GA were also displayed. Despite increasing the numbers of genes, the total amount of features increased dramatically. Due to the resources required to run them and lack of improvements in performance relative to simpler counterparts, GADEN and GADENZ were dropped. The final four FS/R modules retained were PCA, PLS, ADEN, and ADENZ.

9.4.1.3 Comparison of Pattern Recognition Modules

The use of different pattern recognition algorithms did not improve the classification beyond the baseline provided by LDA. Aside from SVMP, the other proposed algorithms did not have a positive mean phi. A potential issue with SVMs was over-fitting, so it may be affected by having a highly imbalanced dataset. As the SVMG and RBF had similar algorithms to each other, the issue of overfitting and difficulty of the data prevented the

sought-after performance gains. When visually depicted (as in Fig. 9.5), the classification boundaries were often overfitted to the training data.

9.4.1.4 Comparison of Classifier Structure Modules

The inclusion of ensembles did not yield the anticipated improvements in the mean phi. Boosting was dropped due to its low performance compared with stacking, bagging, and AdaBoost. Even though AdaBoost had individualized weighting for specific datapoints, it did not surpass the other ensembles. Stacking, however, demonstrated noticeable improvements on the performance metrics. While the mean phi value of stacking with PCA was the same as than previously reported, different FS/R methods did not increase performance in terms of mean phi, sensitivity, or selectivity.

The stacking ensemble directly adjusted the weighting of the meta-features it generated rather than selecting feature indexes, so changing the FS/R method could drop performance (Peiris et al., 2011). Even with the PCA-based stacking ensemble, only the SABIL features achieved the performance benchmark, rather than the SABIS features. The evidence showed that the benchmark performance values became highly situational and reliant upon heavily preprocessed features. Due to the added layer of complexity presented by an ensemble, a single LDA classifier was deemed sufficient for further research.

9.4.1.5 Shortcomings of Traditional Machine Learning Approaches

A problem was the failure of proven machine learning techniques as both single classifiers and ensembles. The lower-than-anticipated performances may be potentially attributable to overfitting, but other factors could be involved. Complications included subject variability, class imbalances, and potentially the loss of temporal information due to the structure of the feature matrix. Even randomizing the order of the epochs did not change the results. The EEG was a noisy representation of brain-state, and many variables required further analysis.

Further investigation into results yielded insights into why the machine learning techniques did not surpass the prior benchmarks. The machine learning techniques investigated have issues distinguishing between close and often overlapping feature spaces, as shown in Fig. 9.7. The top two ADEN features were normalized relative to the highest values for each. The relative closeness of features from 4 microsleeps and 4 alert states are clearly visible.

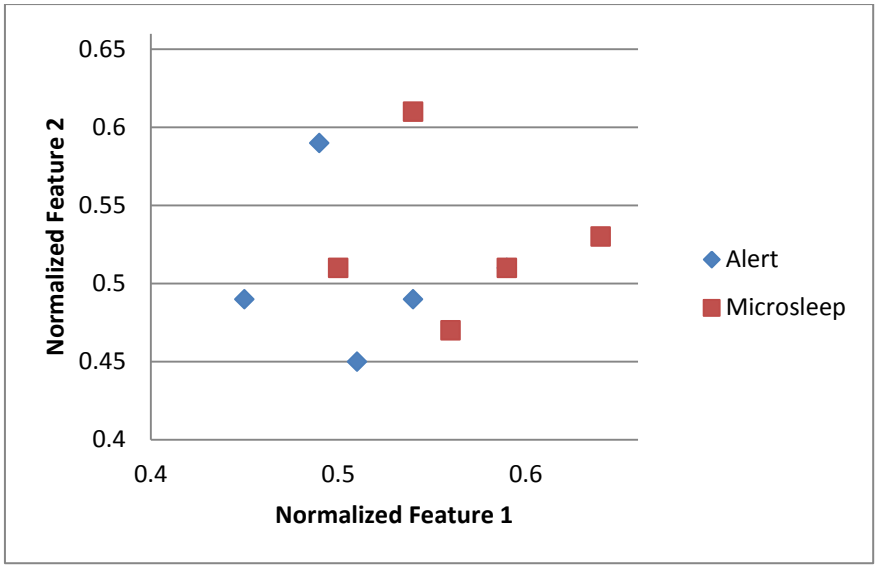

Figure 9.7: Overlapping alert and microsleep group boundaries in SABIL feature set

The findings were relevant to both single classifiers and ensembles (Takenouchi and Eguchi, 2004; Suykens et al., 2002). As discussed in prior work, each of the machine learning techniques had distinct advantages that were thought to provide advantages over the stacking ensemble. However, the results demonstrated that they did not surpass the prior benchmark. In an extended analysis of the literature, certain documented shortcomings of the evaluated configurations were noted (Takenouchi and Eguchi, 2004; Suykens et al., 2002). For example, highly imbalanced data can result in biased classifiers in the case of SVMs, AdaBoost, and RBFs. In AdaBoost, an ensemble comprised of weak classifiers focus on datapoints incorrectly classified by the prior classifier, which gives it a vulnerability to outliers. When the dataset is highly imbalanced and the feature spaces of both categories are close, the entire ensemble becomes poorer at classification (Takenouchi and Eguchi, 2004). For SVMs and RBFs, the clustering algorithms used by the data become prone to overfitting, by compounding incorrect associations between both categories (Suykens et al., 2002).

The ensemble arrangements that performed the lowest included bagging and boosting. While bagging's performance was comparable to AdaBoost, boosting was the lowest performing classifier examined. Boosting shared many of the shortcomings of AdaBoost, but its potential shortcomings included a smaller number of learners and an unweighted majority voting system when compared with AdaBoost (Schapire et al., 2005). Bagging did not perform as well as hoped for. A known limitation of unweighted majority voting bagging was that all classifiers in the ensemble are considered equal. Thus, bagging was hindered by classifiers in the ensemble that were inaccurate, but were still weighted the same as "better" classifiers (Breiman, 1996).

LDA and stacking were less susceptible to these problems for separate reasons. The simplicity of LDA prevented it from overfitting like the SVMs did. The stacking ensemble weighted component classifiers based upon their general performance in data classification (Gandhi et al., 2006). In this way, the influence of more successful classifiers was increased, and the influence of less successful classifiers was reduced. Its component classifiers were LDA, which provided additional robustness to the entire ensemble. Due to these factors, stacking managed to avoid overfitting more successfully than the other ensembles did.

9.4.1.6 Final Selections

After comparing the results, simple and robust system configurations were selected. Among the FS/R modules, PCA, PLS, and the ADEN variants were selected. For pattern recognition, LDA was selected. For classifier structures, single LDA, bagging, stacking, and AdaBoost were selected. However, as a single LDA classifier was the primary component of each ensemble, it was deemed sufficient for most research tasks.

9.5 Summary

The WEKA software toolbox was used to successfully validate ICTOMI's ensemble modules. Afterwards, the most promising modules for further work were selected. In the case of FS/R modules, PCA, PLS, ADEN, and ADENZ were selected. For pattern recognition modules, LDA was used due to its simplicity and reliability on the SABIL and SABIS features. With ensembles, bagging and AdaBoost did not offer discrete benefits over a single LDA classifier, and stacking offered a highly situational benefit dependent on the use of a particular dataset. While an ensemble could improve a performance baseline, a single LDA classifier was deemed sufficient for further research. However, the highly imbalanced nature of the data deserved investigation of advantages in classification which might be obtained by artificial balancing of data, as explored in Chapter 10.

CHAPTER 10. EVALUATION OF CLASS BALANCE VARIATIONS UPON CLASSIFIER PERFORMANCE

10.1 Introduction

With the fundamental modules of ICTOMI implemented and validated, analysis of expert-rated EEG datasets commenced. As previously detailed, different configurations of feature extraction, feature selection/reduction, and classifier structure were investigated with two primary datasets.

The first dataset was Study A (N = 8), in particular, SABIS and SABIL versions. As those feature sets had the highest performance on the earlier results, successful results with the variant training scenarios meant the other feature sets would have a new benchmark to be compared to. If varying the training could not improve the best case scenario, then it was considered unlikely to improve the other datasets.

Due to the imbalance of classes, balanced versions of the SABIS and SABIL feature sets and gold standards were developed. To artificially balance the feature set, events were repeated and a random subset of non-events was deleted until the total composition of the feature set was evenly split between events and non-events. Despite the artificial nature of the balanced feature sets, the possibility that classifiers trained on balanced feature sets and tested on unbalanced feature sets could be an improvement over previous scenarios was considered.

Hypothesis 2: *Artificially altering class balance for training will result in an increase in performance due to removing classifier bias from class imbalance.*

Rationale: Changing the balance of each class for training is a standard technique for dealing with classifier bias (Raudys, 1991).

10.2 Methods

Two system configurations were tested utilizing the SABIS and SABIL features: (1) training and testing on balanced features, and (2) training on balance and testing on unbalanced features. The unbalanced feature results from Section 9.3 were used for comparison.

10.2.1 Feature Selection and Reduction Modules

Six primary feature reduction/selection modules were included alongside an LDA classifier: PCA, ADEN, ADENZ, and PLS. A subset of 10 features or meta-features was used in all other cases.

10.2.2 Pattern Recognition Modules Used

Four pattern recognition modules were studied: LDA, RBF, SVM with Gaussian kernel, and SVM with polynomial kernel. Three LOOCV classifier structures were a single classifier, stacking, boosting, and AdaBoost. Stacking, bagging, and AdaBoost were only investigated with LDA as a classifier. In addition, single LDA classifiers trained on balanced features and testing on unbalanced features were examined. Due to its simplicity and robustness, LDA was the primary classifier used in many cases.

10.3 Results

10.3.1 Training and Testing on Balanced Data

The balanced data was examined alongside the unbalanced data to investigate the effects of altering class balance. Almost universally, the balanced data for each system configuration scored higher than the unbalanced data.

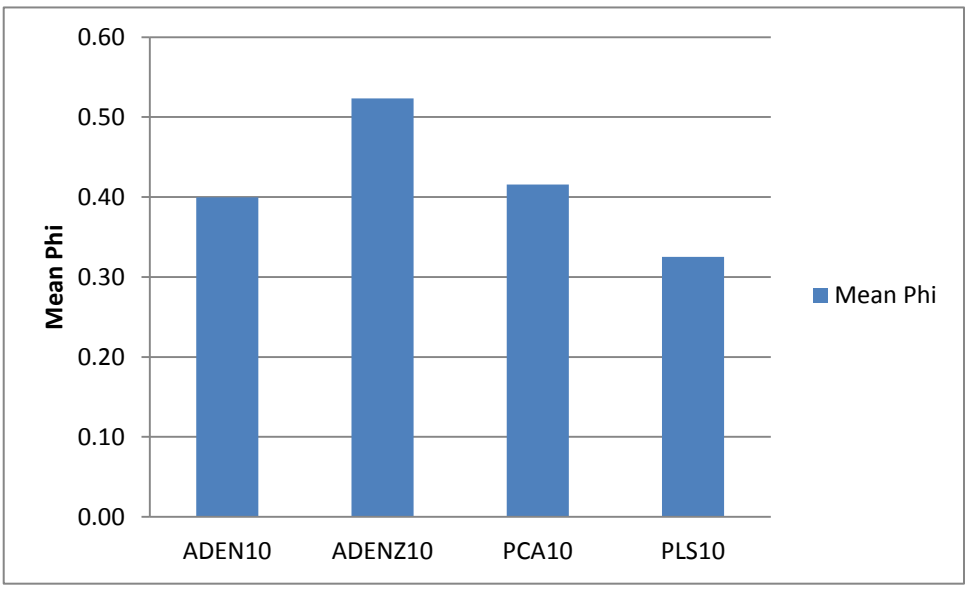

Figure 10.1: Comparison of feature reduction/selection modules with balanced SABIL features

As depicted in Fig. 10.1, PCA performed the highest on the balanced data with the SABIS features, with a phi value of 0.45. ADEN was second with a phi of 0.35. The lowest observed phi coefficient, 0.20, corresponded to PLS.

For the SABIL features, the mean phi values on balanced data were also higher when compared to unbalanced data. The highest mean phi value, 0.52 (0.23-0.71), corresponded to ADENZ. The lowest mean phi value at 10 features was 0.33 (0.22-0.65) with PLS.

10.3.2 Training on Balanced and Testing on Unbalanced

Combined scenarios would involve a classifier trained on a different feature set than it was tested on. The system configuration was training on artificially balanced data and testing on unbalanced data.

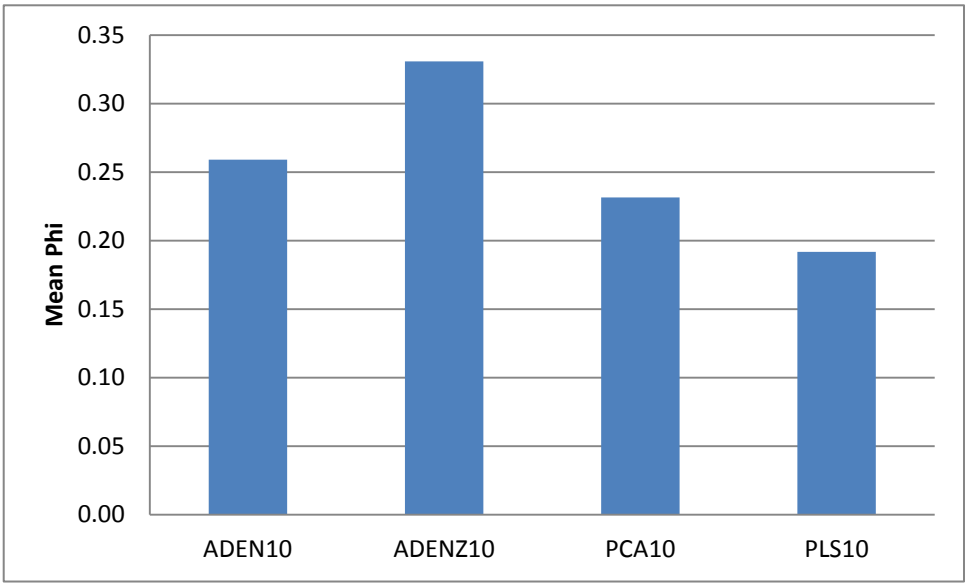

Figure 10.7: Comparison of training on balanced SABIL features and testing on unbalanced data

As detailed in Fig. 10.2, PCA scored the highest mean phi with the SABIS data with a single LDA classifier, while ADEN and PLS scored close to each other. With the SABIL features, the highest value corresponded to 10 ADENZ features with a mean phi value of 0.33. The second highest mean phi value was 0.26 with 10 ADEN features. The mean phi values seen for both feature sets did not surpass the prior benchmark.

10.3.3 Comparisons

A comparison is shown in Fig. 10.3, where the unbalanced case resulted in a phi of 0.33 with 10 ADENZ features with the SABIL feature set.

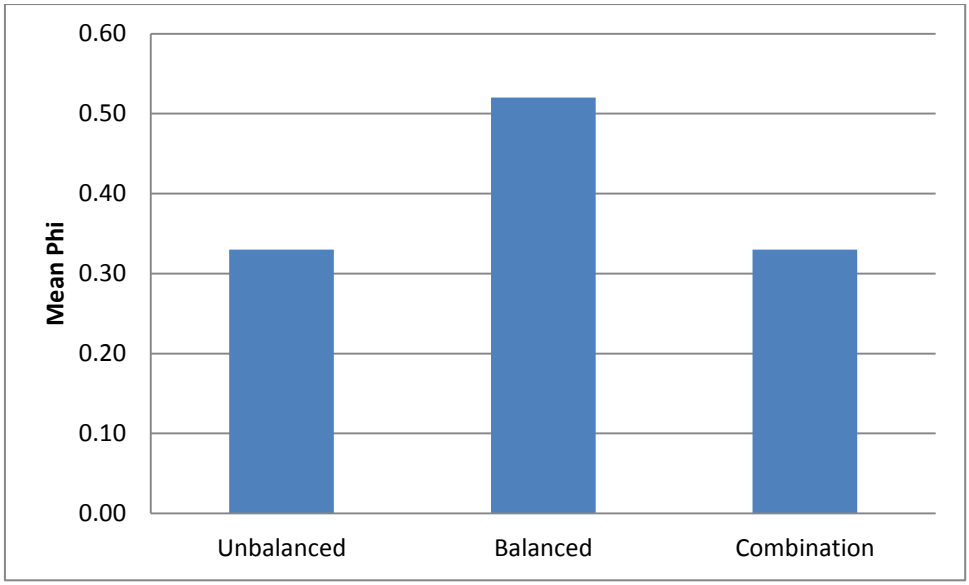

Figure 10.3: Phi values of SABIL results with 10 ADENZ features using: a) Unbalanced, b) Balanced, and c) Combination

The phi value on the unbalanced case was the same as the combination case, where the classifier was trained on balanced and tested on unbalanced. The only increase occurred during the balanced case.

10.4 Discussion

The SABIS and SABIL feature sets were evaluated with the intention of seeking to exceed benchmark performances from prior work, such as the mean phi value of 0.39 (Davidson et al., 2007; Peiris et al., 2011). Both feature sets were considered to be the most informative data regarding spectral features and brain-state, but potentially hindered by their class imbalance. The primary findings can broadly be grouped into those from unbalanced data, those from balanced data, and those from unconventional training scenarios.

The use of balanced data for training and testing dramatically boosted the mean phi correlation, but is an unrealistic scenario. Microsleeps are highly infrequent events in real life, but the balanced data presented a hypothetical scenario where commonality apparently made them easier to detect. In the results of balanced artificial event data, artificially increasing observations of an imbalanced class did not increase the mean phi correlation. It may be that the nature of the artificial event feature sets to account for such results, but increasing the number of events in training may successfully boost performance on unbalanced data.

Training on balanced data but testing on unbalanced data did not result in increased performance, but it did not have highly adverse results. Instead, mean phi values decreased

slightly with respect to training and testing on unbalanced data. Again, PCA, ADEN, and ADENZ corresponded to the highest values, while PLS corresponded to the lowest. In addition, training on pruned balanced data and testing on unbalanced data was undertaken. The use of pruned data did not increase the phi correlation coefficient beyond what had been witnessed in prior scenarios. Training on balanced data and testing on unbalanced data provided little benefit, so it would be excluded.

The following step would be to examine and compare variant system configurations with the same data. As LDA proved more than sufficient for classification in most cases, SVM and RBF pattern classification modules would be dropped. While training on a balanced dataset might not have improved mean phi values or other performance metrics, the concept of training on an optimized subset of data remained.

The use of balanced data would also be dropped, as the classifier outputs depended upon the individual subject tested. It was thought that training on balanced data would improve pattern recognition modules and ensembles due to known issues with highly imbalanced, small-sample data. Despite the investigation into shortcomings of the traditional machine learning techniques, the greatest performance variances were attributable to individual subjects raising or lowering the total mean phi.

For the investigated system configurations, the third and fourth subjects in Study A corresponded to the lowest testing performance. Further investigation into individual subjects in Study A was absent from the literature, so intra-subject variance relative to inter-subject variance required further evaluation.

10.5 Summary

The SABIS and SABIL feature sets were exhaustively covered using a variety of feature reduction/selection and pattern recognition techniques in both balanced and unbalanced cases. An artificially balanced version was used for training, which offered no quantifiable benefits over training and testing on unbalanced data. The concept of optimizing the training data warranted further investigation, especially by a thorough analysis of the individual subjects, as explained in Chapter 11.

CHAPTER 11. INVESTIGATION OF SUBJECT VARIABILITY ON CLASSIFIER PERFORMANCE

11.1 Introduction

Previous microsleep detection results were averaged from a range of individual performance values. Certain subjects had consistently low values across multiple system configurations with the same feature set. Even across the Study A feature sets, certain subjects consistently scored higher or lower than the average phi value. Understanding the reasons for successful individual subject classification was thought to provide insights into improving microsleep detection.

Preprocessing of the data was deemed to be a key step in the process due to the many changes in the process. Study A had more variant feature sets based upon the range of preprocessing techniques applied to it. As such, a systematic investigation and comparison into their performances was undertaken. For example, if two individual feature sets each possessed a drastically different range and mean, then the effects of changing preprocessing on individual subjects would be examined.

A starting point for the research was the direct comparison of variant preprocessing steps. After investigation of the SABIS and SABIL feature sets, analysis was performed on the SARUS and SABUS feature sets lacking ICA preprocessing and artefact pruning. Without the preprocessing steps performed on the SABIL and SABIS feature sets, the SARUS and SABUS feature sets were thought to resemble a more realistic classification problem suitable for an online microsleep detection system. The benchmark results were based upon data that had undergone ICA and artefact pruning for testing, so the effect of testing with the artefact sessions included was unknown (Peiris et al., 2011).

A comparison of feature sets with different methods of preprocessing could provide additional insights. The comparison between referential and bipolar EEG features was not covered in prior work (Peiris et al., 2011). While the benchmark results were achieved with features from bipolar EEG, the ability of referential EEG to achieve similar results was examined. If both achieved comparable results on the same system configuration, the information would be relevant during the implementation of an EEG headset.

Study C required closer examination due to having been examined in few system configurations. When analysed with PCA and a single LDA classifier in LOOCV, the highest

mean phi value was -0.03 (-0.31-0.09). The spread of values was far lower than the results for Study A feature sets, so examining each individual subject was thought to provide additional insight into the classification difficulties.

Hypothesis 3: *There is a positive correlation between classifier performance and mean microsleep duration.*

Rationale: Studying the individual subjects in Studies A and C could allow for additional insight as to the classifier results. If common factors are found for subjects that perform exceptionally well or poorly when tested for microsleep detection, adjustments could be made to the system.

11.2 Methods

11.2.1 Intra-Subject Examination

Determining an optimal set of training data required the establishment and characterization of a standard metric. Individual differences between subjects meant that generalizing across the experimental population was difficult. No prior metric existed for estimating the signal to noise ratio for an individual subject. Due to variable conditions, including electrode connections, individual variations, or other factors, certain individuals' microsleeps may be undetectable by sensors. It was hypothesized that by identifying and removing these subjects from the training data, that a classifier could be optimized. The phi correlation contained the most relevant information, so it was the primary metric for the research.

In order to test "undetectability," within-subject classification was used. A within-subject classification problem was considered to be a "best case scenario" for classifiers, due to other variables being constrained to one person. Two-fold cross-validation was used after randomly partitioning the data into two approximately equal sets of events and non-events. An averaged sum of the phi correlation from twofold cross-validation on LDA with three feature reduction/selection modules (PCA, ADEN, and PLS) was taken as the primary score.

Study C only had one session per subject, so the possibility of changes between sessions did not need to be considered. While Study A had two sessions, within-subjects tests were performed with both sessions concatenated together. Since little difference was found in performance between Study A's two sessions and the prior convention of combining both sessions, the values presented are from the inclusion and testing on observations from both sessions per subject.

It was considered that a mean phi value ≤ 0.10 corresponded to random guessing. The value served as an initial threshold to determine if feature data from a particular subject could be classified successfully. If the mean phi from a single subject's within-subject classification task dipped below the threshold, the subject was considered to have undetectable events. The mean phi values were arranged on a spectrum from least to greatest, with the threshold later increased to a mean phi value of 0.15.

11.2.1.1 Personalized Microsleep Detection

A related topic to within-subject microsleep classification was the possibility of personalizing a microsleep detection system for an individual. An innate limitation with the datasets examined is the number of EEG recordings for each subject, as fewer sessions means less microsleeps and a shorter EEG duration. Study A had only a pair of sessions per subject, and Study C only had one session per subject.

While a dataset with a larger number of sessions per subject would have been preferable, the potential for a personalized microsleep detector trained on only a subset of the subject's total EEG was considered. A key hindrance to this avenue of research was the small number of subjects. Due to the already small number of subjects, removing a single subject could easily cause shortfalls in a classifier's ability to generalize across a wider pool of individuals.

Further research into the potential effects of variance in impedance and other factors between sessions would need to be conducted. The potential for evaluating personalized microsleep detectors would be limited only to training and testing on all data from a single subject. While this was not ideal, it would determine how many subjects in Studies A and C that intra-subject classification was viable for.

Analysis was carried out using a variation of LOOCV. For each subject, features from all sessions were concatenated into a single matrix. The dimensions of the matrix were spectral features from all channels by the total number of 2-s epochs. Half of the epochs in the matrix were selected at random. The remaining epochs were used for training a classifier, before testing on the previously removed data. The training and testing data were then reversed, and the phi values from each case were averaged together.

The three system configurations used were based upon an LDA classifier in conjunction with ADEN, PLS, or PCA. All systems used 10 features (in the case of ADEN) or meta-features (in the case of PCA and PLS) for the analysis. For each subject, the resultant phis of ADEN, PCA, and PLS were averaged together for a final score. It was expected that

within-subject values would be higher than phi values from a LOOCV configuration, due to a within-subject classifier only being evaluated on one subject after being trained on data from the same individual. Due to the limited sessions for each subject in Studies A and C, a single bad session could easily consign a subject to being undetectable.

11.2.2 Management of Undetectable Subjects

Managing subjects deemed undetectable were handled in different ways. The most direct method was standard LOOCV, systematically excluding each subjects in a feature set and then training a single classifier or ensemble upon the remainder. Following this, subjects were excluded from training and testing based upon the mean phi value from the within-subject classification task. Following this, a classifier was trained on all feature data, save those features of the excluded individuals. However, subjects excluded from the training dataset were still included in the testing dataset. Based upon these results, the threshold was adjusted upwards to 0.15 and the previous steps repeated.

The exclusion of subjects from small datasets was considered to result in reduced generalization. Following this, a "mixed" scenario was evaluated, in which data from all subjects in a dataset was randomly recombined into L=5 blocks, so that L-fold cross-validation would occur with random subsets of data from all subjects. Each block would be treated as a synthetic subject, so a classifier would be trained on features from 4 blocks and tested on the previously unseen features of the fifth. LOOCV was performed to ensure each block was used as the test data. If certain subjects were undetectable, then the randomized recombination of features into blocks would provide an alternative method for evaluating individual configurations.

Likewise, machine learning scenarios were often investigated for comparison in a temporally independent context. If the Study A or Study C feature sets had information that could be successfully generalized across all subjects in the feature set, the validity of using that feature set for analysis would be reinforced. For feature selection methods like ADEN and ADENZ, the specific indices of features calculated from the best mixed performances were retroactively applied to standard cross-validation to determine if feature set performance was boosted. In addition, the common features selected across all subjects by the ADEN and ADENZ were examined for further insights.

An adaptive system able to adjust parameters for an unseen subject's data was not specifically investigated. However, bagging was investigated, where component classifiers were trained on randomized blocks of data recombined from training subjects. Conventional

LOOCV was used as a control when compared with bagging and mixing data, as depicted in Fig. 11.1. For both bagging and mixing, the same value of *L* used was 5 for cross-validation.

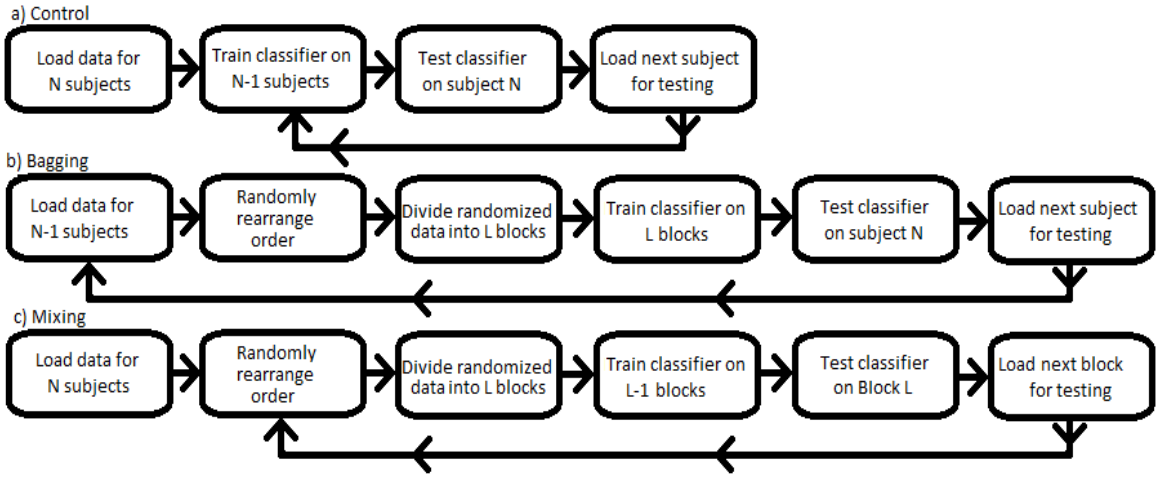

Figure 11.1: Comparative schematic of system training and testing methodologies with a) Control, b) Bagging, and c) Mixing

Any differences could illustrate the effects of removing subjects from training while dealing with small datasets containing highly imbalanced classes. Due to generalization, the mixed case was considered to give comparable mean phi values for each case of LOOCV. As the specific testing subject could determine the performance results, bagging and the conventional control scenarios were hypothesized to be more subject-dependent than the mixing case.

11.2.3 Investigated Configurations

The system configurations investigated were PCA, ADEN, ADENZ, and PLS for FS/R, with classifiers including a single LDA classifier, an LDA-based stacking ensemble, and an AdaBoost ensemble of 30 weak learners with the initial number of 10 features per system configuration. If the ensemble configuration did not notably increase performance, they would be dropped from further investigation so that single classifier configurations could be studied more thoroughly by varying the amount of features. Five feature sets were tested in total: SABIL, SABIS, SARUS, SABUS, and SCRIS. It was hypothesized that while a classifier may lose generalization through the exclusion of undetectable subjects from training, a possible trade-off in performance had to be investigated. Conversely, a consistent mean phi returned from randomly mixed data higher than the subject exclusion-based LOOCV mean phi would highlight the susceptibility of a linear classifier performance due to the exclusion of individuals with small, highly imbalanced datasets.

11.3 Results

The SABIL feature set, SABIS feature set, SARUS feature set, the SABUS feature set, and SCRIS feature set were tested under several system configurations. The first set of results emerged from standard LOOCV using a single LDA classifier, stacking ensemble, and AdaBoost with 30 weak learners. Following this, certain system configurations were revisited utilizing the removal of undetectable subjects. Finally, alternative training and testing approaches were followed, with training and testing divided by the "undetectability" criterion.

11.3.1 Cross-Validation Results

LOOCV was used with a single LDA classifier, stacking, and AdaBoost. A single LDA classifier was used as the principal preliminary test for each feature set. On Study A, discrepancies were noted between performance on the "raw" (SARUS and SABUS) and SABIS feature sets. The highest mean phi for the SABIL features was ADENZ with 0.33 (0.12-0.52) on the SABIL features. For the SABIS features, the highest mean phi with PCA was 0.27 (0.0-0.51), while being 0.15 (0.02 to 0.31) with the SARUS features and 0.18 (0.04-0.34) on the SABUS features. ADENZ scored a mean phi of 0.27 (0.00-0.51) on the SABIS Study A feature set and a 0.33 (0.12-0.52) on the SABIL feature set, while the highest mean phi performance on either raw feature set corresponded to ADEN with a value of 0.21 (0.05-0.36). The lowest mean phi value on Study A corresponded to PLS on the SABIL feature set with 0.19, the SABIS feature set with 0.18 (0.02-0.36), the SARUS with 0.13 (-0.09-0.27), and SABUS feature set with 0.11 (-0.15-0.32).

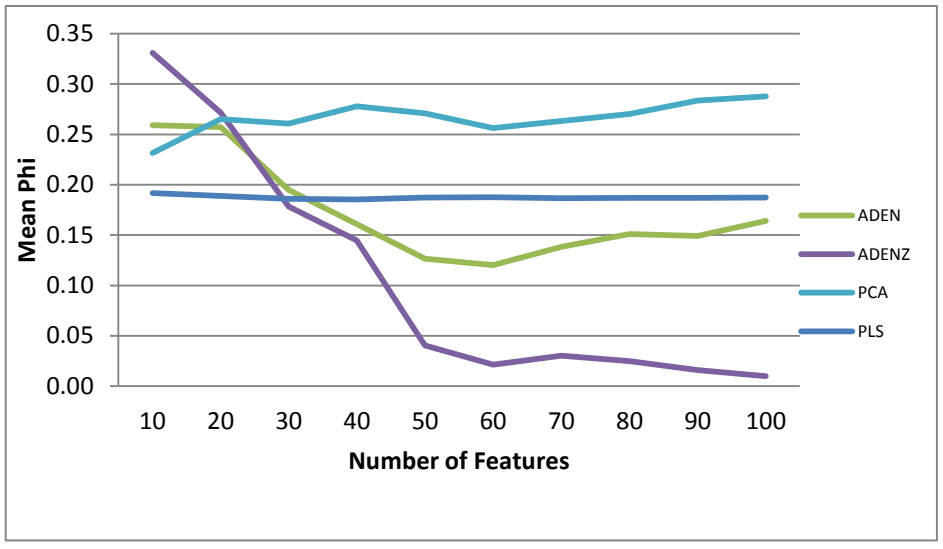

Figure 11.2: Results of SABIL features on a single LDA classifier

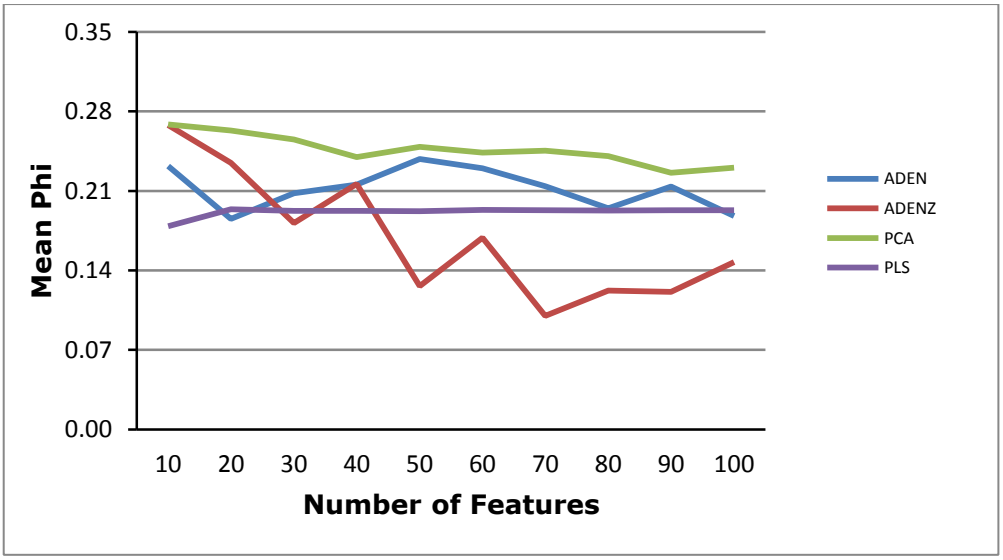

Figure 11.8: Results of SABIS features on a single LDA classifier

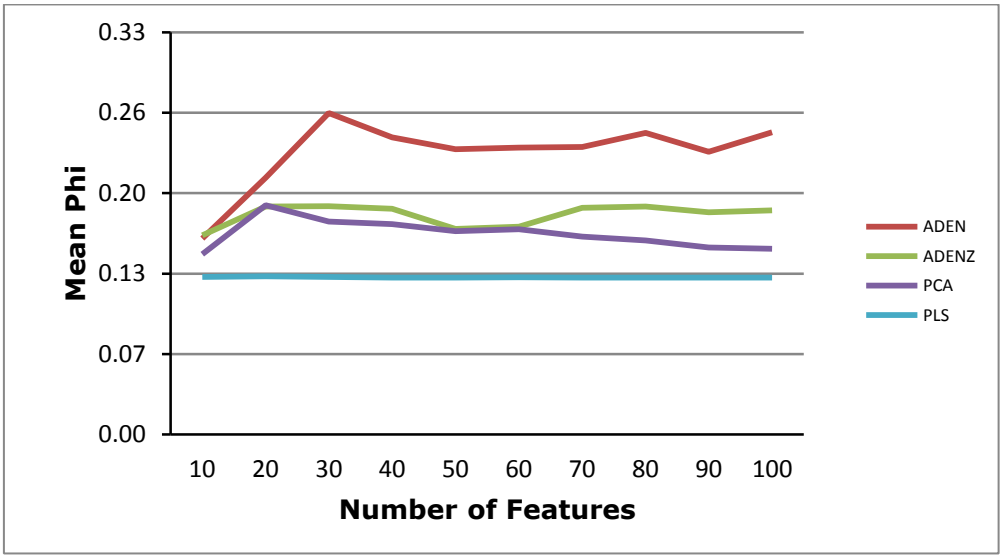

Figure 11.4: Results of SARUS features on a single LDA classifier

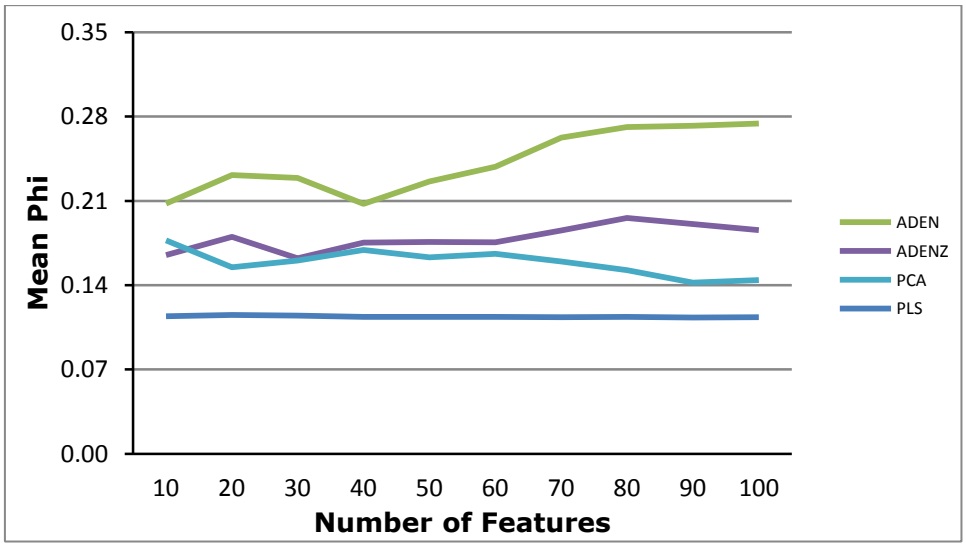

Figure 11.5: Results of SABUS features on a single LDA classifier

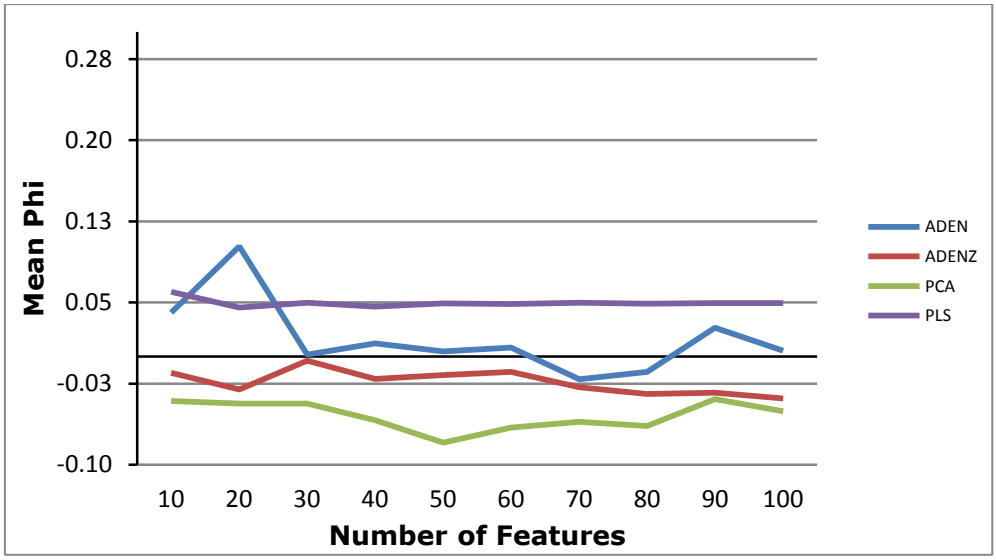

Figure 11.6: Results of SCRIS features on a single LDA classifier

The mean phi performance of the SCRIS feature set was lower than Study A performance. On SCRIS features, PCA scored the lowest mean phi at -0.04 (-0.31-0.09), while PLS corresponded to the highest mean phi at 0.06 (-0.11-0.21). ADEN and ADENZ were between the two in performance, corresponding to a mean phi of 0.04 (-0.02-0.12) and 0.06 (-0.07-0.27).

Following the initial evaluation of FS/R methods, the classifier structure was changed from a single classifier to stacking and AdaBoost ensembles. The highest mean phi performance with an ensemble was with the SABIL features was 0.40 (0.13-0.66) with 150 PCs on the stacking ensemble. The lowest mean phi value with the stacking ensemble on the SABIL features was 0.16 (0.01-0.37) with 10 PCs. The highest mean phi performance with the SABIS feature set with ADENZ at 0.23 (0.06-0.56). The lowest value on the same feature set corresponded to PCA with a mean phi of −0.16 (-0.39- −0.01). Between the SARUS and SABUS feature sets, the highest stacking score corresponded to a mean phi of 0.21 for both ADEN (0.06-0.36) and ADENZ (0.04-0.38) with the SABUS features. On SCRIS, the highest value corresponded to PCA with a mean phi of 0.10 (-0.07-0.38). The lowest mean phi value on SCRIS was ADEN with 0.00 (-0.03-0.04).

With AdaBoost, the highest mean phi values of 0.20 (0.02-0.54) came from ADEN with the SABIL features. For the SABIS features with AdaBoost, the highest mean phi again corresponded with PCA at 0.19 (-0.03-0.50). The lowest value with the SABIS features came from ADEN with a mean phi value of 0.14 (0.00-0.41). On the SABUS and the SARUS feature sets, the highest value was PLS with the SARUS features, scoring a mean phi of 0.06 (-0.01-0.14). On SCRIS, the highest mean phi value was PLS with 0.09 (-0.05-0.46). The

lowest score for SCRIS came from PCA with a mean phi of -0.01 (-0.09-0.10). Due to a lack of performance relative to single classifiers, ensemble configurations were not thoroughly investigated for the following research.

ADEN and ADENZ were used to select the features with the highest average distances between microsleep and alert states, as shown in Table 11.1.

Table 11.1: Top SABIL features selected from across all subjects using a) ADEN and b) ADENZ

a) Top ADEN Features

Type	SP	SP	SP	SP	SP	SP
Weight	1.00	0.94	0.93	0.89	0.86	0.85
Band	Alpha	Alpha	Beta	Gamma	Alpha	Beta
Channel	P3-O1	T4-T6	Fp2-F8	T5-O1	P3-O1	Fp1-F3

b) Top ADENZ Features

Type	NSP	PR	NSP	NSP	PR	PR
Weight	1.00	0.95	0.92	0.90	0.88	0.87
Band	Alpha	Theta/Beta	Alpha	Beta	Alpha/Beta	Alpha/Beta
Channel	P3-O1	C3-P3	F3-C3	Fp2-F8	C4-P4	T6-O2

SP: Spectral Power NSP: Normalized Spectral Power

PR: Power Ratio

Each feature was assigned a weighting value based on the maximum normalized distance between features. Each algorithm selected separate features due to different methods of normalizing distances, although the difference in classification performance was negligible. Features from the alpha spectral band were the ones with the highest distances computed by ADEN and ADENZ across all subjects. For individual subjects, the features with the greatest difference between states represented delta, theta, alpha, beta, and gamma bands.

11.3.2 Removal of Undetectable Subjects

The removal of the so-called “undetectable” subjects changed the results before and after adjusting the threshold. The results from the within-subject (WS) mean phi tests were different across both Study A and Study C, as depicted in Table 11.2.

Table 11.2: Within-subject mean phi values for Study A (a) and Study C (b)

a) Study A

Subject	804	809	810	811	814	817	819	820
WS Phi	0.33	0.50	0.02	0.14	0.57	0.22	0.29	0.36

b) Study C

Subject	203	207	208	210	211	213	214	216	217	220
WS Phi	0.33	0.05	0.22	0.22	0.53	0.50	-0.03	-0.03	0.12	-0.25

At the initial mean phi threshold of 0.10, four subjects (207, 214, 216, and 220) were excluded from Study C and one subject (810) was excluded from Study A. When the mean phi threshold was raised to 0.15, a single subject was excluded from both Study A (811) and Study C (217). The within-subject performance was compared with individual performance on LOOCV in Fig. 11.7.

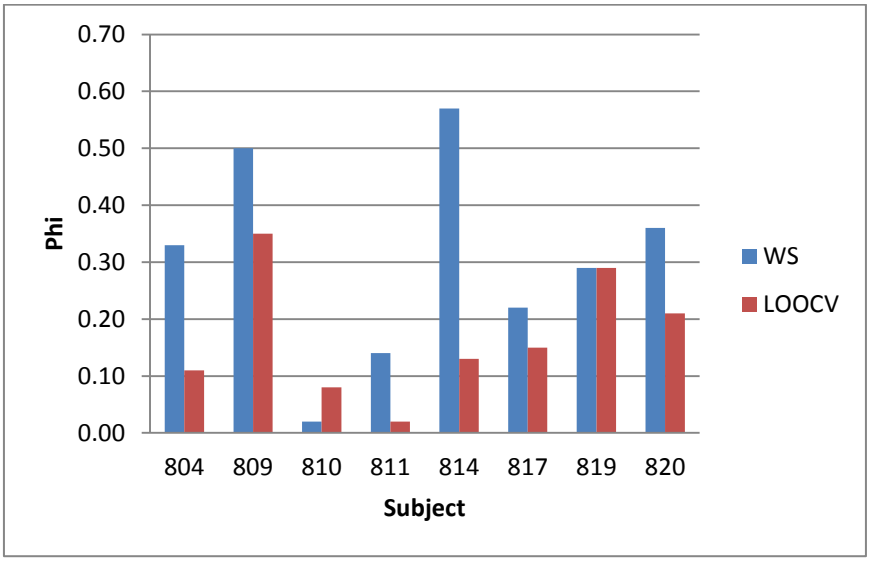

Figure 11.7: Comparison of within-subject phi and individual subject performance on LOOCV with PCA on a single LDA classifier

As expected, Fig. 11.7 demonstrates that even though a subject had a higher WS phi, it does not always translate into a high performance when that subject is used as a test subject in LOOCV. While within-subject phi values were higher than LOOCV scores on average, an

individual subject would occasionally perform higher than the within-subject mean phi. While Subject 810 scored slightly higher on LOOCV than WS, the difference was not meaningful. Both scores were below a phi value of 0.10, as the classifier had difficult classifying events from Subject 810. With Study A, the exclusion of data had effects even with a single classifier. The highest results corresponded to the SABIL features. The highest mean phi rate was 0.44 (0.28-0.64) with 150 PCs. The second highest mean phi value corresponded to the SABIS features, in which both PCA and ADENZ achieved a mean phi of 0.31 (0.00-0.51). The value was an improvement over the single classifier mean phi value of 0.27 (0.00-0.51) in both cases. For the SARUS features, highest mean phi performance dropped from 0.16 (0.02-0.33) to 0.12 (0.00-0.21) in the case of ADEN. However, performance with ADENZ dropped less, from a mean phi of 0.16 (0.01-0.38) to 0.14 (-0.03-0.31). For Study C, the highest mean phi performance without removing subjects was 0.06 (0.00-0.30) in the case of PLS. After removing the first undetectable subjects from the pool, the highest mean phi values corresponded to 0.10 in the case of both ADEN and PLS.

The largest increases in performance corresponded to the SABIL features, while the other datasets registered only incremental changes after applying the final threshold for mean WS phi (> 0.15). With the SABIL features, the highest mean phi value corresponded to 0.46 (0.27-0.66) with 150 PCs. With the SABIS features, the highest mean phi value corresponded to 0.37 (0.05-0.52) with PCA following the removal of two subjects. The highest mean value for Study C was 0.15 (-0.02-0.56) with PLS.

11.3.3 Mixed Training Scenarios

The mixed data evaluation involved the random recombination of data from all subjects in a feature set into 5 blocks. Each block was equivalent to a synthetic subject. One block would be excluded for testing and the remainder used for training a single LDA classifier. Mixing was compared with bagging to determine the effects of entirely leaving a subject out of the training data.

With bagging, each of the four Study A feature sets performed optimally under different circumstances. At 10 features, the highest mean phi value corresponded to the SABUS features at 0.23 (0.07-0.38) with PCA, even higher than the SABIS features with a mean phi of 0.21 (0.03-0.46). The maximum mean phi for the SABIL features occurred at 10 ADEN features at 0.30 (-0.03-0.53). The highest mean phi occurred with 100 ADEN features on the SARUS features at 0.25 (0.04-0.40). For SCRIS, the maximum mean phi corresponded to PCA with 10 features at 0.12 (-0.04-0.38).

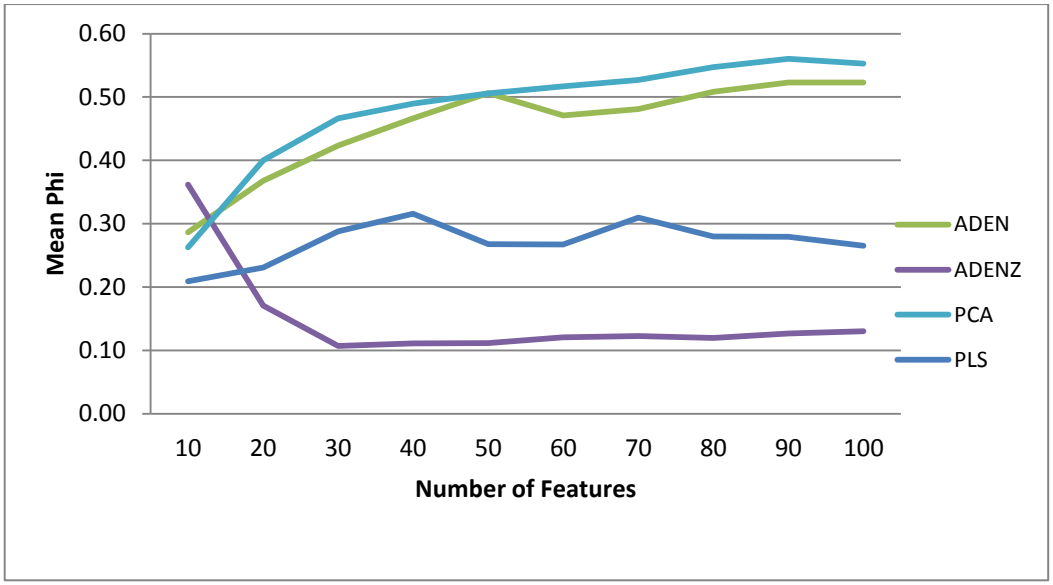

Figure 11.8: Results of SABIL mixed features on a single LDA classifier

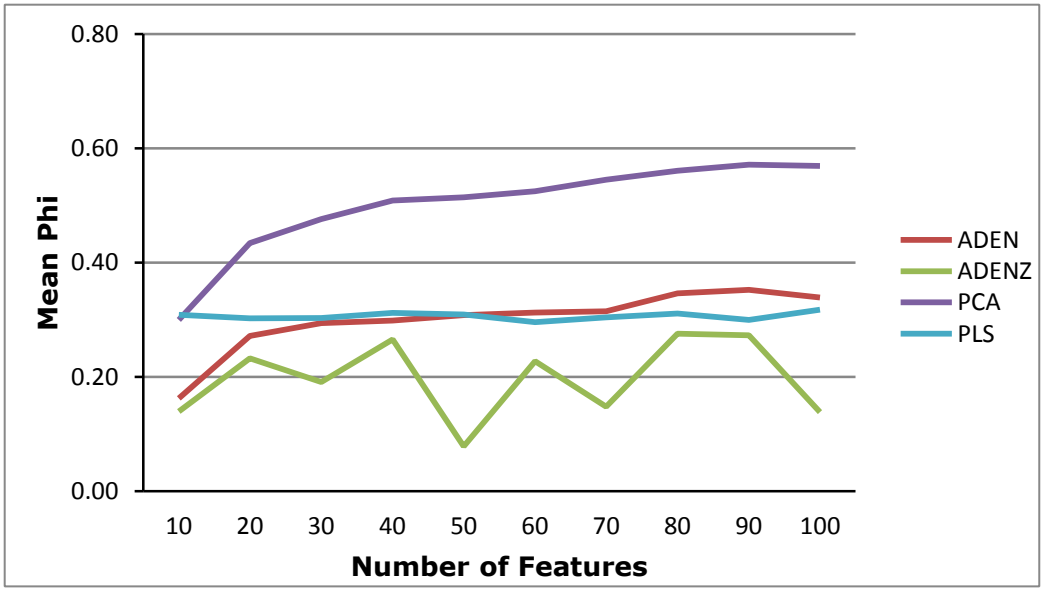

Figure 11.9: Results of SABIS mixed features on a single LDA classifier

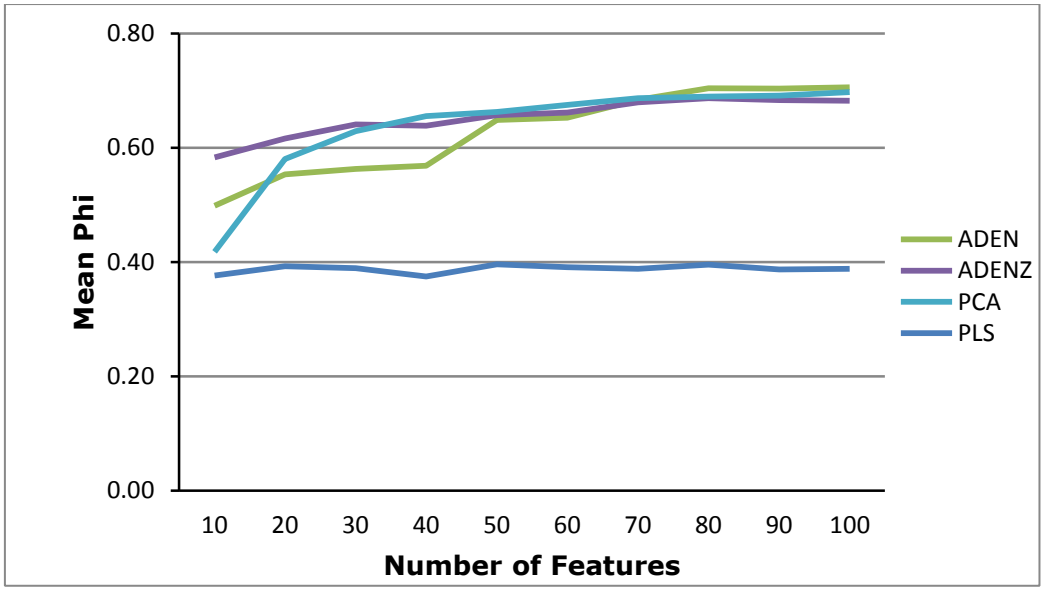

Figure 11.10: Results of SCRIS mixed features on a single LDA classifier

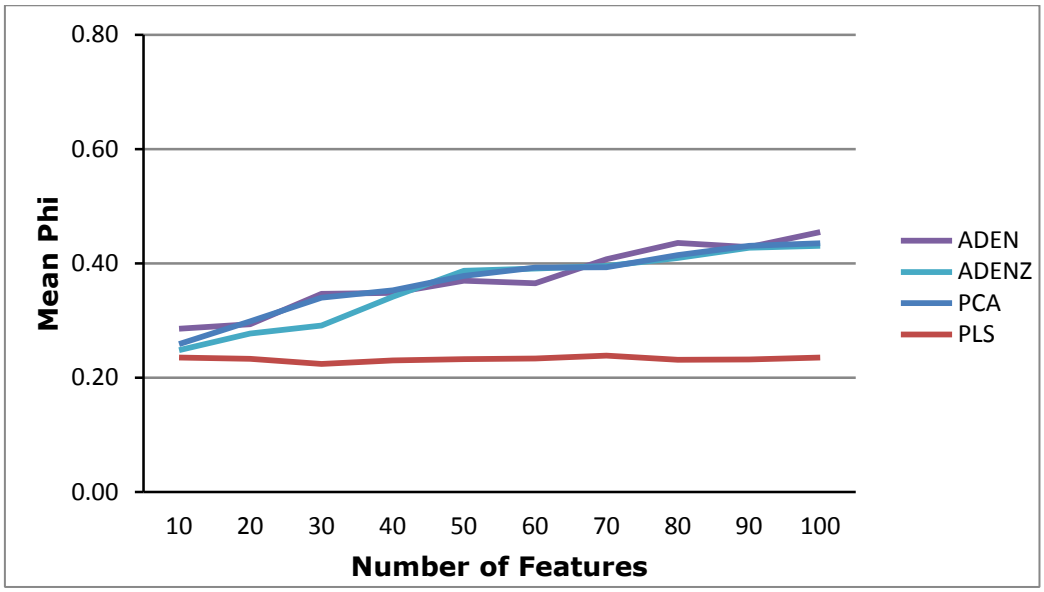

Figure 11.9: Results of SARUS mixed features on a single LDA classifier

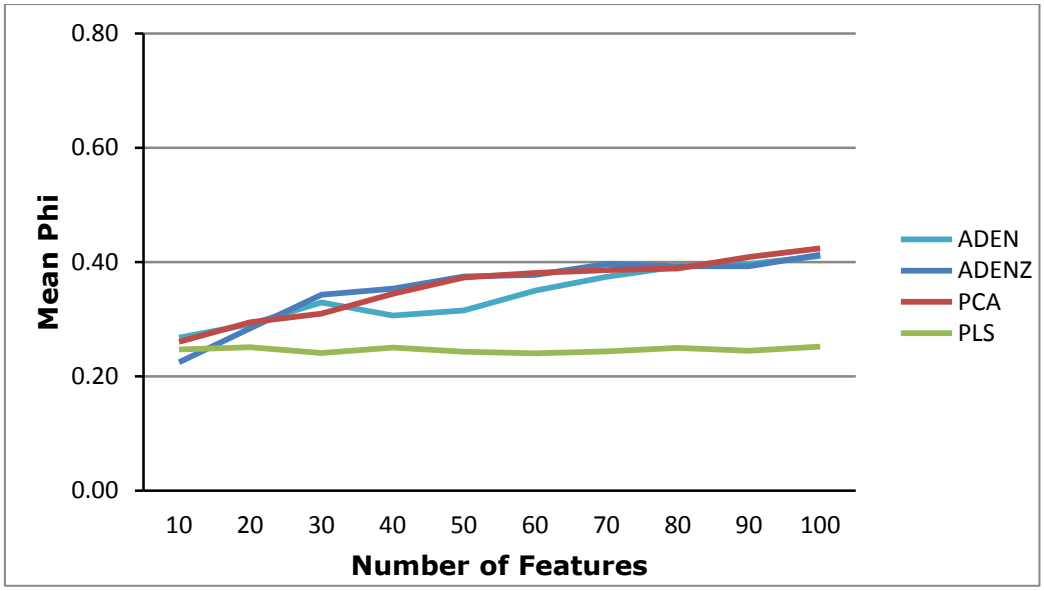

Figure 11.10: Results of SABUS mixed features on a single LDA classifier

With the mixed feature data, mean phi values were higher than the conventional LOOCV scenarios as the number of features was increased. The highest mean phi value for the SABIL data was 0.56 (0.55-0.57) at 90 PCs. The highest mean phi on the SABIS features corresponded to PCA with 90 features at 0.57 (0.55-0.59). On the SARUS feature set, the highest mean phi was associated with 100 ADEN features at 0.46 (0.45-0.46). On the SABUS feature set, the maximum mean phi corresponded to PCA with 100 features at 0.42 (0.41-0.44). However, the highest mean value corresponded to the Study C features at 0.71 (0.70-0.71) at 100 ADEN features.

11.4 Discussion

Given the variety of approaches tried, certain implications regarding potential microsleep "undetectability" were drawn. The comprehensive battery of tests registered only incremental and trivial increases in mean phi performance when ensemble systems were added. The removal of subjects demonstrated that both Study A and C contained individuals with microsleeps that could not be classified by conventional LOOCV systems.

The mixed feature analysis required a larger number of features to reach their maximum phi values. ADEN, ADENZ, and PCA demonstrated a gradual trend to increase phi value as more features were included. PLS remained relatively constant in phi value as the number of features increased. PLS overfitting was considered a possibility as to the reason.

The mixed approach to the data drastically improved mean phi performance on each feature set by allowing the classifiers to generalize across the entire experimental population.

However, a real-time microsleep detection system would be unable to mix data together into synthetic subjects, but other information was gleaned from the practice. Under certain conditions, even linear classifiers were sufficient for imbalanced datasets of noisy spectral features.

11.4.1 Feature Set Interpretation

The feature sets themselves exhibited dramatically different performances, even within the same study. The dramatic difference in performance between the SABIS, SABIL, SARUS, and SABUS feature sets was attributed to a combination of factors. The SABIS and SABIL feature sets had automated artefact pruning and ICA applied to them, while the "raw" (SARUS and SABUS) feature sets did not. As such, many of the harder-to-classify or noisier segments of the data present in the "raw" features were not present in the SABIS features.

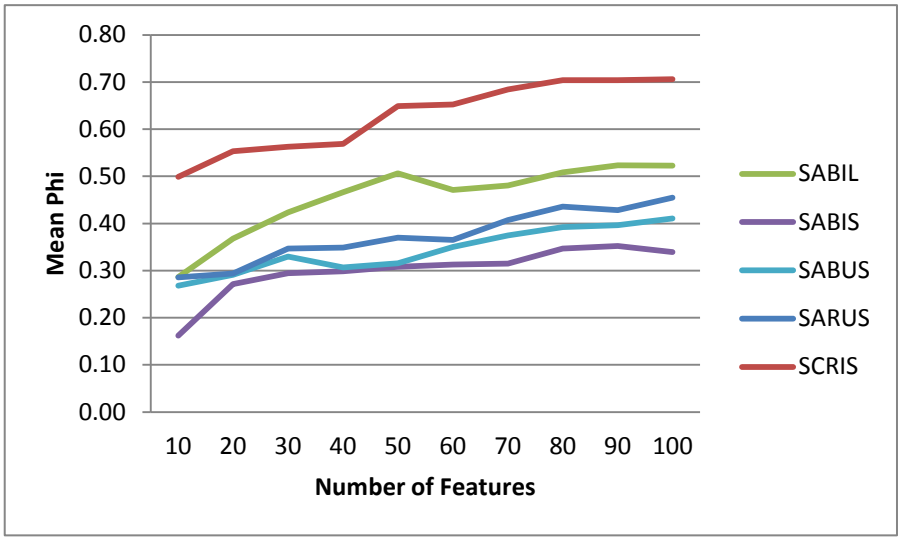

Figure 11.11: Comparative performance of ADEN with a single LDA classifier on mixed major feature sets

The SARUS and SABUS feature sets often exhibited similarities in performance. While both often resulted in mean phi values lower than the SABIS feature set, mean phi performances might closely approximate the performance of a microsleep classification system upon "realistic" features. Despite earlier work (Peiris et al., 2011) using bipolar EEG channels in the belief they provided a cleaner signal than referential EEG, the SARUS EEG feature set achieved a higher mean phi correlation than the SABUS EEG in a number of circumstances. The differences in most cases were not significant, but the ability for referential EEG features to achieve the maximum mean phi correlation could imply that conversion to bipolar may not necessarily be required for successful microsleep detection.

11.4.1.1 Survey and Interpretation of Top Selected Features

A comparison of the feature indices selected by ADEN and ADENZ for each of the Study A feature sets was substantially different. The divergences in selected indices only increased upon the removal of the undetectable subjects, demonstrating their effect upon feature selection even after similar preprocessing steps for each respective feature set.

For each feature set, ADEN and ADENZ computed a subset of features that yielded the highest performances across subjects. In general, they corresponded to changes on the alpha and theta bands, but the specific features varied for each feature set. ADEN and ADENZ often found different results. For the SABIL feature set, the top features calculated using ADEN included the alpha band spectral power from the bipolar channels T4-T6 and P3-O1. Using ADENZ, the top SABIL features across subjects included the normalized alpha band power on channels F3-C3 and P3-O1. Individual subjects had a broader range of features, including gamma band power and normalized beta power. The prevalence of activity near the motor cortex and midbrain was consistent, providing insight into the brain-states relevant to microsleeps from both studies.

Based upon prior literature, changes in EEG before and during microsleep were observed on the alpha and theta bands (Davidson et al., 2007; Poudel et al., 2010). Due to this, it was anticipated that features derived from the theta and alpha bands would appear more often than the others. The appearance of features corresponding to the beta and gamma bands amongst the top features indicates these frequency bands contain information regarding brain-states relevant to microsleep detection.

Further analysis on brain-states relevant to microsleep detection was limited due to the nature of the available data. The results were found on spectral features derived from noisy EEG, and the gold standard used in the classification task was based on an incomplete knowledge of microsleeps (Poudel et al., 2010). Due to these limitations, analysis was conducted into previously overlooked subject factors.

11.4.1.2 Analysis of Additional Subject Factors

Demographic data of individual subjects in Study A and C had not been considered during prior microsleep research. In addition, the knowledge of microsleeps was greater in Study C than Study A. Since half of Study C's individual subjects possessed low WS phi values, possible secondary and demographic factors were examined more closely. Hence, Study C was examined more closely than Study A, as secondary factors associated with low mean WS phi values were examined.

In Study C, a significant difference was found between the distributions of undetectable and "detectable" subjects with regards to microsleep duration. Longer microsleeps corresponded to a within-subject mean phi greater than 0.15. While excluding subjects greatly increased mean phi values, the loss of data represented a potential loss in the ability to generalize. As the Study C data and the SCRIS features were entirely referential, additional evidence was supplied that bipolar conversion may not necessarily provide benefits for microsleep detection.

An unexpected aspect of Study C was possible correlations between mean WS phi values and mean microsleep duration. In particular, mean microsleep duration greater than 3 s corresponded directly with mean WS phi values of greater than 0.15. Subjects with mean durations <3 s corresponded to mean WS values below the threshold at 0.15. A t-test was used to investigate the interaction of factors, returning a significant p <0.001. Even a *t*-test on mean microsleep duration returned a significant p = 0.0054. Mean WS phi values also showed significant differences with a p = 0.0152. The correlation coefficient was 0.67, providing evidence of an interaction between the two variables.

In Study A, the mean duration of 1.5 s served as the threshold between the detectable and undetectable subjects and low WS phi values. ANOVA returned a significant p = 0.002. A *t*-test on mean duration resulted in a significant p = 0.005. A *t*-test on mean WS phi resulted in a significant p = 0.002. The correlation value for mean duration and mean WS phi was 0.57, slightly less than 0.67 for Study C. While Study A and Study C had different "threshold durations," the 3 s value from Study C was preferred due to Study C's extensive documentation on types of lapses. In both Study A and Study C, the longer duration of microsleeps likely allowed for the classifier to better identify relevant spectral changes in an individual.

11.4.2 Personalized Microsleep Detection

The potential for personalizing a microsleep detector was hindered by the low number of sessions, but the preliminary results were promising. Due to high WS phi values for most of Study A, the potential for a personal microsleep detector exists. While Study A only has two sessions per subject, it is of note that WS values were higher than Study C's WS values. If a simple LDA classifier can perform up to 0.57 on a single subject, then a personalized classifier may be even higher when trained on enough data. A classifier that works well on one subject may not work well on another.

The results demonstrated that intra-subject classification could present an alternative method for gauging the effectiveness of microsleep classification systems, in contrast with LOOCV. LOOCV, used elsewhere, can vary greatly on the quality of a single subject's features. When a system has been personalized, it can achieve substantially higher results with even a rudimentary classification system. For personalizing a microsleep detector, a greater number of sessions per subject would be preferable to a single session from many subjects, due to the ability of a classifier to generalize across multiple sessions over time. However, the low number of sessions per subject hindered further work on this particular research avenue. Another drawback to personalized microsleep detection is the necessity of an "external" gold standard for each patient. An individual would need to come into a lab for an initial calibration, although other sensors may eliminate the need for this in the future. The need for a verifiable gold standard to initially calibrate a system for an individual placed a large burden on further research in this direction.

11.4.3 Training Method Interpretation

A potential limitation was the exclusive use of linear classifiers to the exclusion of all others. Ensemble systems, such as stacking, potentially over-fitted in certain cases. By using an ensemble system, an additional layer of complexity was added to an already complex classification system. The linear classifier used in the case of the single classifier and as the basic unit of an ensemble might have been insufficient to find meaningful patterns in the spectral features. Alternative structures, such as probabilistic, adaptive systems, or deep learning, might provide an answer.

Another issue with the study was the exclusive reliance upon subject-based LOOCV. A classifier's performance is largely dependent upon the ability to draw meaningful patterns of correlation on between classes of a subject's spectral features, but being tested upon features of "lesser" quality than the training set would result in a low classification accuracy.

A low within-subject phi on a subject corresponded to a successful predictor of low performance when a subject was used for testing. For "undetectable" subjects, the classifier is unable to correctly identify microsleeps and alert states.

An alternative to subject-based cross-validation investigated was training and testing upon randomly-selected blocks of data from all subjects in a dataset. Even a linear system improved in performance at generalizing between classes when given access to a larger cross-section of data.

The mixed feature data achieved superior performance by establishing subject-independent generalization. A realistic microsleep detection system would not (initially) be calibrated for a new user, so benefits of the system do not apply to cases with subject exclusion. While the increase in performance was clearly visible in the four Study A feature sets, the increase in the SCRIS feature set was dramatic. Jumping from a mean phi of 0.01 (-0.09-0.29) to 0.71 (0.70-0.71) with ADEN with 100 features means that potentially, enough information exists in linearly discernable spectral features to account for absent channels. The feature indices selected by ADEN and ADENZ may be applied back to standard LOOCV to see if performance increased. Additionally, correlations between specific spectral bands and electrode channels could be studied in greater detail. However, such a practice can only apply to feature selection techniques (e.g., ADEN and ADENZ), rather than meta-feature generation techniques (e.g., PCA and PLS).

11.5 Summary

The removal of subjects scoring below a threshold value of 0.10 on intra-subject cross-validation resulted in performance increases for both Study A and Study C. Further raising the threshold had little effect. The initial case resulted in an average value of 26.5% of the subjects being classified as undetectable. The small experimental population hindered attempts to find a precise value. Given the substantial percentage of undetectable individuals, EEG-based microsleep detectors should utilize other sensors (e.g., video and accelerometers) as backup measures. Dynamic weighting of features may also prove useful in optimizing a classifier for an individual. However, the feature selected using the mixed method may be applied in a more conventional way. Additionally, the effects of preprocessing can radically affect the feature selection, as can changes in the training data. Potentially changing the training data to a more generalized format might prompt the selection of better features across a wider range of subjects. Alternatively, a larger number of sessions with one individual might allow for personalization of microsleep detection. The connection between mean microsleep duration and mean WS phi may additionally warrant further investigation, given that low values for both corresponded directly to the undetectable subjects.

In order to reduce the execution time of both training and testing, an FS/R method based on mixed subject analysis was conducted (Chapter 12).

CHAPTER 12. MIXED-SUBJECT FEATURE SELECTION TRIALS

12.1 Introduction

An issue with LOOCV was changes in performance due to the subject variability. The mixed data approach demonstrated that despite subject variability, enough information was still present between microsleeps and alert states to achieve phi values as high as 0.71 (0.70-0.71) with 100 ADEN features on the SCRIS features. Due to the structure of the LOOCV, ADEN would select different features due to subject variability, potentially missing useful information uncovered by the mixed data approach. By using the specific feature indexes from the mixed subjects data approach, it was hoped the LOOCV results could be improved.

The potential improvement for LOOCV results was thought to be substantial. For example, 100 ADEN features with mixed 5-fold LOOCV on the SCRIS features achieved a mean phi of 0.71 (0.70-0.71), over random guessing when standard LOOCV was used. This indicated that ADEN, with a sufficiently generalized selection of features, is able to identify microsleeps even across subjects with a highly imbalanced feature set. If this success could be transferred to standard LOOCV, then it was hypothesized that performance would be increased for supervised learning techniques. The approach was named mixed-subject feature selection (MISFETS).

An additional application of MISFETS was as a preprocessing technique. By selecting a high enough number of features, enough information could be retained by a smaller subset of data. The potential to increase speed of execution was considered, especially in conjunction with other methods of FS/R.

Hypothesis 4: *Selection of an optimal set of spectral features via MISFETS will boost microsleep detection performance.*

Rationale: MISFETS potentially allowed an optimal subset of features to be selected from each feature set, narrowing down a large volume of data. The resulting feature set could have higher performance metrics than the standard feature sets due to holding the most relevant information.

12.2 Methods

The use of MISFETS originated with ADEN. The specific ADEN and ADENZ indices calculated during the mixed cross-validation were compared with indices calculated during standard LOOCV. Two sets of all unique feature indices were taken from the mixed cases with ADEN and ADENZ, which were used to generate the indices used by MISFETS.

Following this, two subsets of the respective features were selected. All features excluded by the subset were deleted for that particular case. Standard LOOCV was then performed using the remaining features for both training and testing. The methodology and origin of feature index subsets was varied.

The first method of selecting feature indices was simply to increase the subset of included features by order. A list of the ADEN or ADENZ feature indices was loaded, and an increasing number of them were included, ranging from 10 to 100 at increments of 10. The scenarios were evaluated using the SABIL, SABIS, SARUS, and SABUS features, in addition to the SCRIS features. For comparison, a random subset of 10 indices from the ADEN and ADENZ were selected without any genetic-algorithm-based enhancement. The random selections of ADEN and ADENZ indices were referred to as RADEN and RADENZ. The purpose of including RADEN was to see if random selection of an already narrow feature set alone would yield comparable performance with a high performance indicating potential to explore a given configuration with GADEN.

Additionally, each of the four feature sets examined was reduced in size to a subset of its original size, before being inserted into any classifier system. The sole criteria for a feature index being included in the subset was belonging to one or both of the list of indices corresponding to the mixed ADEN or ADENZ cases. Following this, standard feature reduction/selection methodologies (ADEN, ADENZ, PLS, and PCA) were applied to conventional LOOCV. If the mixed case truly generated an optimal subset of features, then performance on the "abridged" feature sets was expected to increase. In addition, certain correlations and observations for spectral bands and channels were made due to each feature corresponding directly to spectral information from a particular channel.

Finally, an alternative to the abridged feature sets was used as a comparison. Training was conducted with only one subject, but testing would involve the others in the feature set. The purpose of the "limited training" cross-validation was to compare the effects of limiting the training size and testing on a wider experimental population. It was performed with ADEN and ADENZ with and without use of MISFETS features. Scores were compared with each other and against standard LOOCV results.

It was believed that the effect of training a single subject would later affect the features that were selected. If successful linear separation between microsleep and alert state features could be successfully applied on one subject, then the features successfully generalized to an entire population. However, if repeatable separation did not exist, then the features selected might be non-optimal. As such, the average results for an entire feature set

might depend upon the “quality” of the features. The WS mean phi was used as a benchmark as far as ranking feature quality, so the SABIS feature set was expected to have the highest mean phi results. The SCRIS feature set was expected to have the lowest mean phi results.

A limited exploration of GADEN and GADENZ was undertaken, using a limit of 100 of the highest ranked ADEN features from MISFETS with 20 offspring over three generations. The number of features to optimize was started at 10 and increased by 10 each iteration until 100 was reached. The process was applied with all five feature sets, with expectations that the SABIL feature set would have the maximum mean phi. In addition, it was hypothesized that GADEN would score higher than ADEN with a similar number of features due to the potential gains of orthogonally selected features over presumably selecting collinear ones.

12.3 Results

Six FS/R methods were investigated across all four feature sets. While the particulars varied, ADEN and ADENZ were used in all cases.

12.3.1 Mixed Feature Selection Approach

Mixed feature selection utilized ADEN, ADENZ, RADEN, and RADENZ. The highest mean phi value for the SABIS features originated from 0.28 (0.01-0.52) with 10 RADENZ features. The highest value for the SABIL feature set was 0.33 (0.12-0.52) with 10 ADENZ features. For the SARUS feature set, the best mean phi correlation was 0.26 (0.03-0.41) with 20 RADEN features. For the SABUS feature set, the highest mean phi was 0.24 (0.10-0.48) from 90 ADEN features. The highest mean phi value for SCRIS was 0.00 (0.00-0.00) with 10 ADEN features.

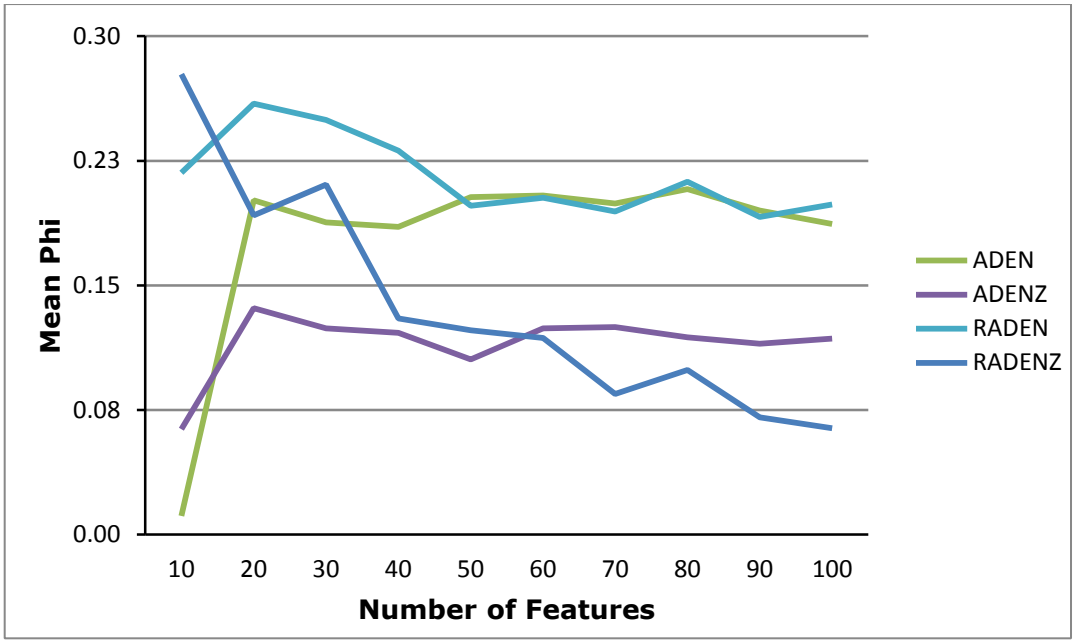

Figure 12.1: Results of SABIS features with MISFETS on a single LDA classifier

12.3.2 Abridged Combination Approach

The four abridged feature sets were utilized with ADEN, PCA, ADENZ, and PLS plus an LDA classifier. The highest value for the SABIS features was ADENZ with 30 features and mean phi of 0.27 (0.04-0.49). The highest mean phi for the SABIL features was 0.33 (0.12-0.53) with 10 ADENZ features. The highest mean phi for the SARUS features was 0.27 (0.03-0.44). The highest mean phi value for the SABUS features was ADEN with 70 features at 0.26 (0.00-0.47). The highest mean phi value for SCRIS was PCA with 30 features at 0.05 (-0.09-0.34).

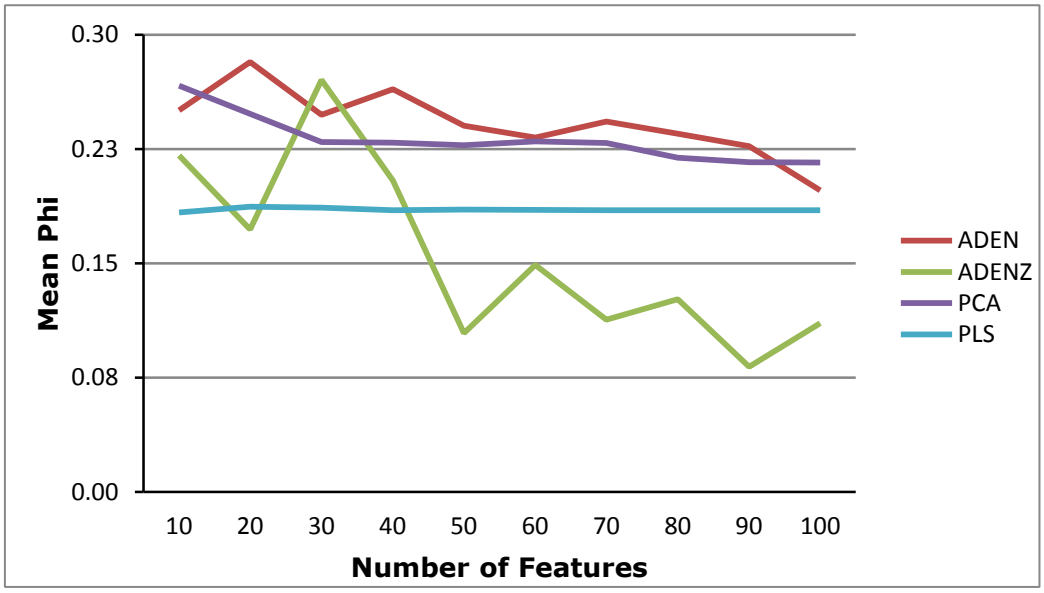

Figure 12.2: Results of SABIS features on a single LDA classifier

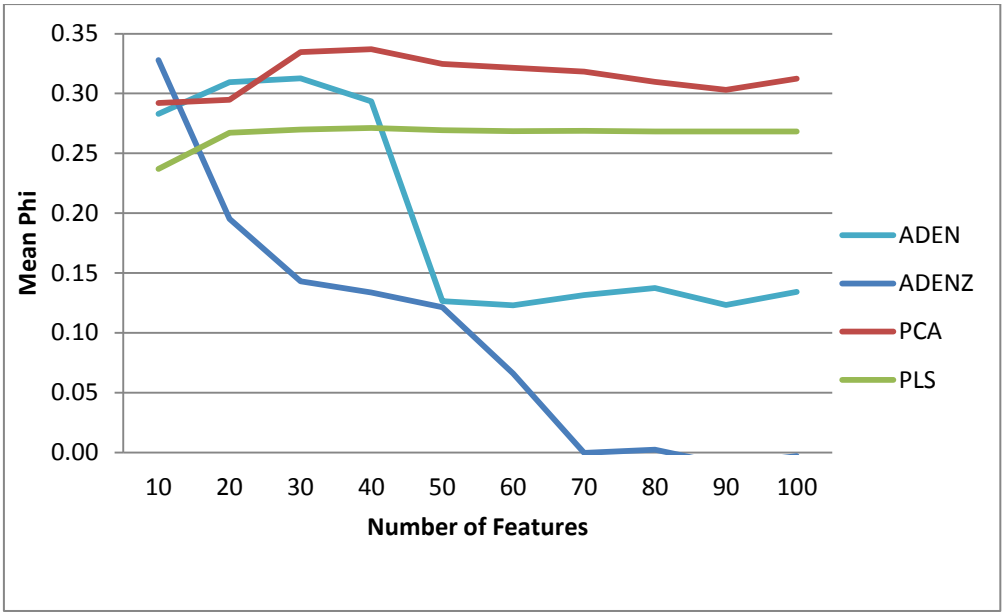

Figure 12.3: Results of SABIL features on a single LDA classifier

No configuration surpassed the prior mean phi record of 0.28 with 10 RADENZ features.

12.3.3 Limited Subject Learning

Due to the innate issues with subject-based LOOCV, MISFETS was used with another method of cross-validation. The highest mean phi value on the SABIS feature set was 0.27 (0.00-0.51) with 10 ADENZ features. The highest mean phi value for the SARUS feature set was 0.26 (0.05-0.43) with 30 ADENZ features. The highest mean phi value for the SABUS feature set was 0.27 (0.02-0.57) with 100 ADEN features. The highest mean phi for the SABIL features was 0.13 (0.02-0.25) with 10 ADENZ features. The highest mean phi value from the SCRIS feature set was 0.10 (0.00-0.32) with 20 ADEN features.

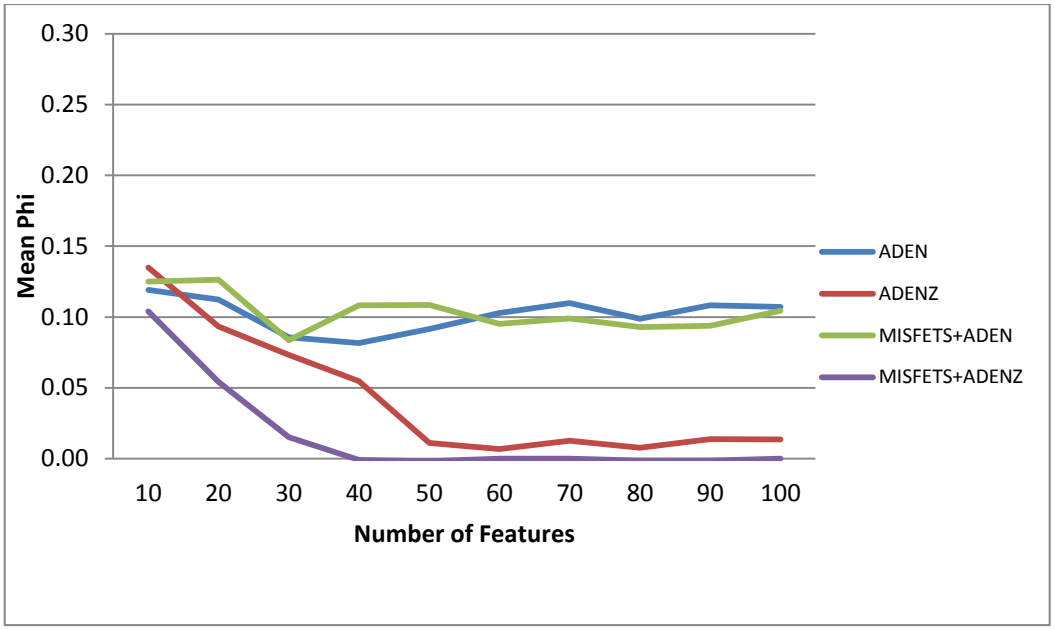

Figure 12.12: Results of SABIL features on a Single LDA classifier trained on one subject

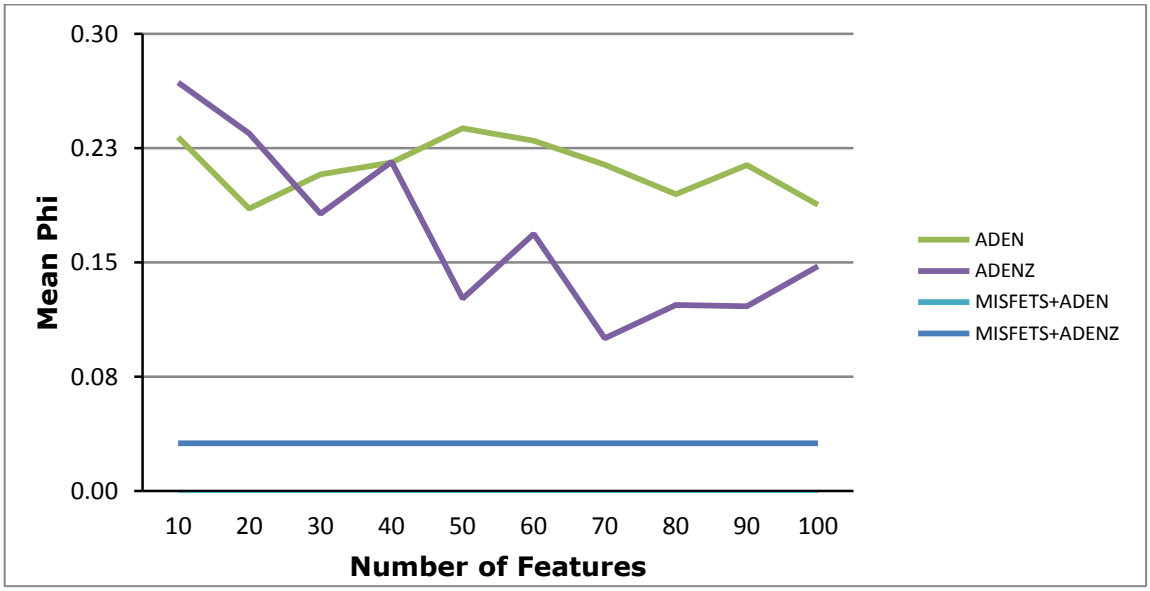

Figure 12.5: Results of SABIS features on a single LDA classifier trained on one subject

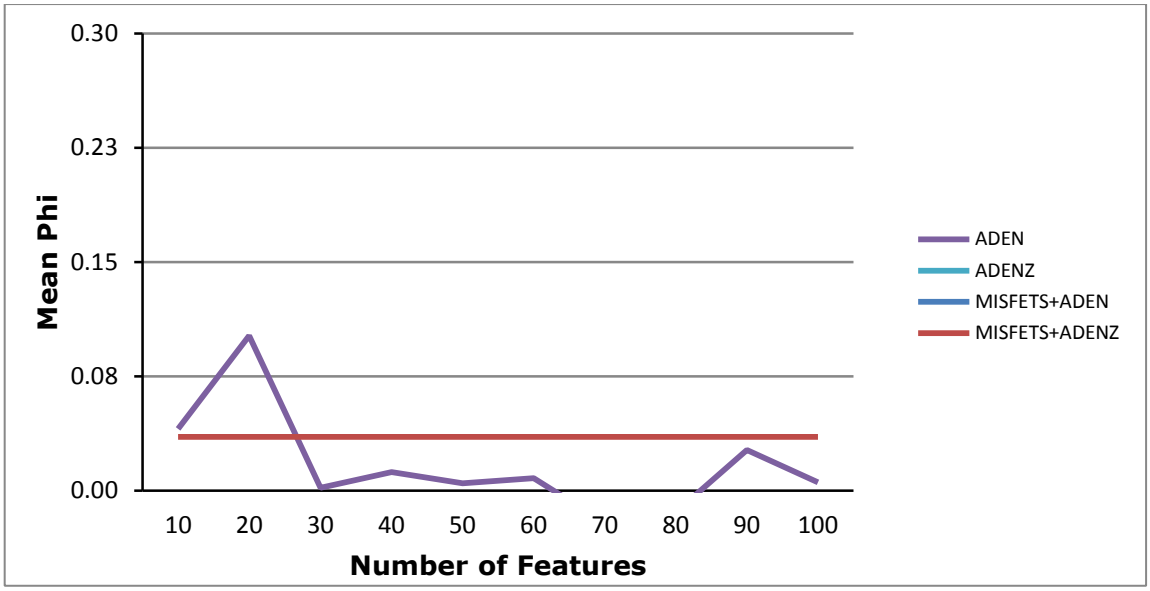

Figure 12.6: Results of SCRIS features on a single LDA classifier trained on one subject

Classifier performance on the SABUS features from Study A equalled the SABIS features in several cases.

12.3.4 GADEN Results

Following of GADEN with 10 features, the results were compared with standard ADEN and MISFETS. The initial number of features to optimize was 10, selected from the top 100 ADEN or ADENZ features from the entire pool of MISFETS features. The highest mean phi for the SABIL dataset was GADENZ with 0.33 (0.12-0.52). The highest performance was achieved by the SABIS feature set with a mean phi of 0.26 (-0.01-0.57). The highest mean phi for the SABUS feature set was GADENZ with 0.20 (0.05-0.34). The highest mean phi for the SARUS feature set was 0.19 (0.01-0.35) with GADEN. The lowest of the maximum mean phi values came with the SCRIS feature set at 0.00 (-0.02-0.01).

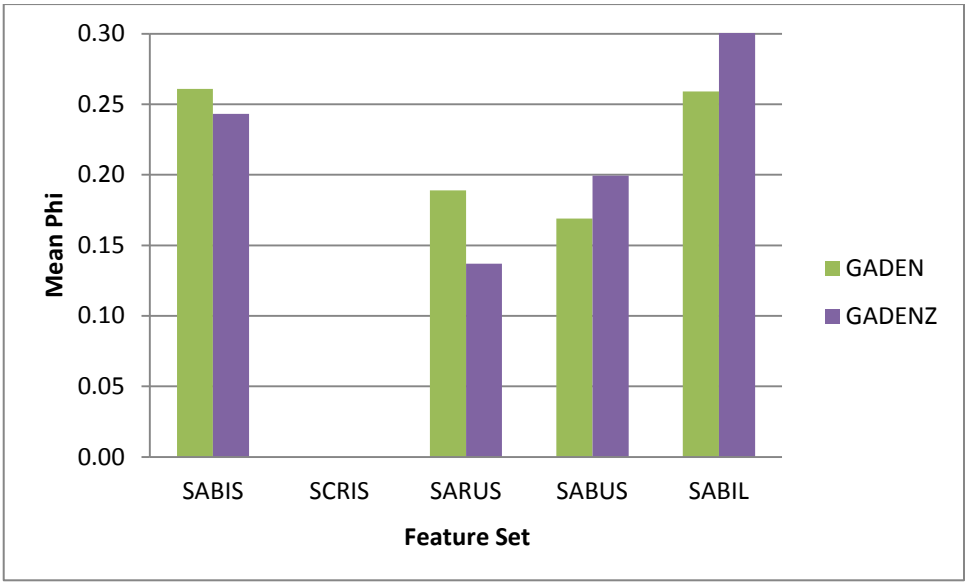

Figure 12.7: All feature sets results with GADEN$_{10}$+MISFETS on a single LDA classifier

The highest GADENZ mean phi value of 0.33 did not surpass the highest mean phi performance by other means, 0.40 with the stacking ensemble.

12.4 Discussion

A thorough analysis of ADEN-permutations demonstrated the circumstances that can result in high performance. The specific ADEN and ADENZ indices calculated during the mixed cross-validation were compared with standard LOOCV. Taking feature indices from MISFETS did not cause the increases in performance hoped for. The resulting increases in performance were incremental and trivial, or equivalent to prior results. The results indicated that for a linear classifier, whether trained on several subjects or one, the specific feature indices were not sufficient to guarantee increases in performance. While the use of another pattern recognition module might have had different results, LDA demonstrated its fundamental limitations.

The inclusion of randomized feature selection algorithms (i.e., RADEN and GADEN) boosted performance higher than standard ADEN and ADENZ in some cases, although certain maximum mean phi correlations did correspond to RADEN or RADENZ. The increases in mean phi performance were unreliable and erratic when compared with standard techniques, so any potential gains from orthogonal feature selection did not give the anticipated increases in performance. In the case of Study A, feature selection between the feature sets varied greatly. For example, the MISFETS features selected from the SABIS, SARUS, and SABUS were primarily different. In SCRIS, many of the MISFETS features

from the generalized feature set were absent in certain subjects. These complications resulted in inferior performance, as witnessed with the SABIL features.

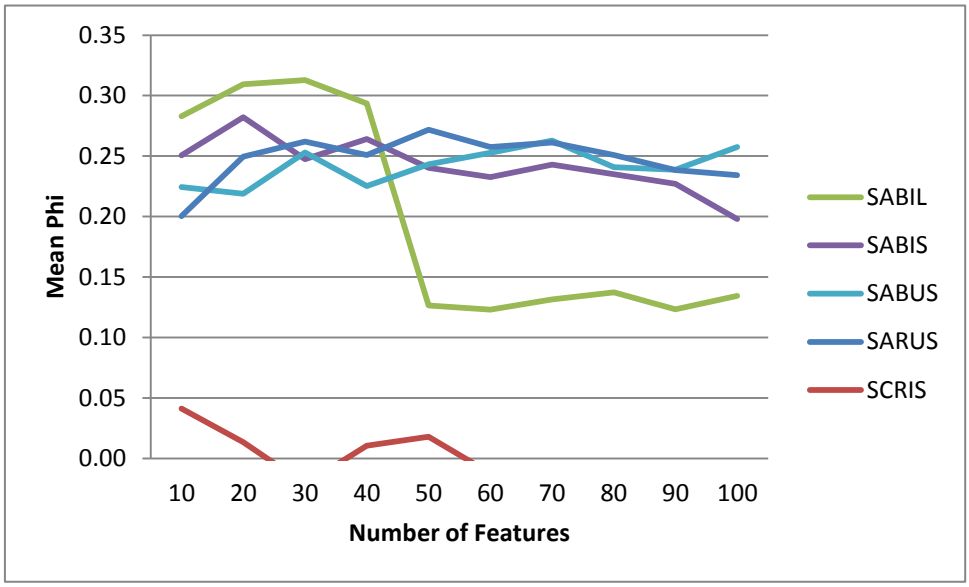

Figure 12.8: Comparative performance of ADEN with a single LDA classifier on major feature sets with abridged features

The reduction of feature sets in size using MISFETS and then performing other FS/R techniques upon the abridged features did not improve performance for arguably similar reasons.

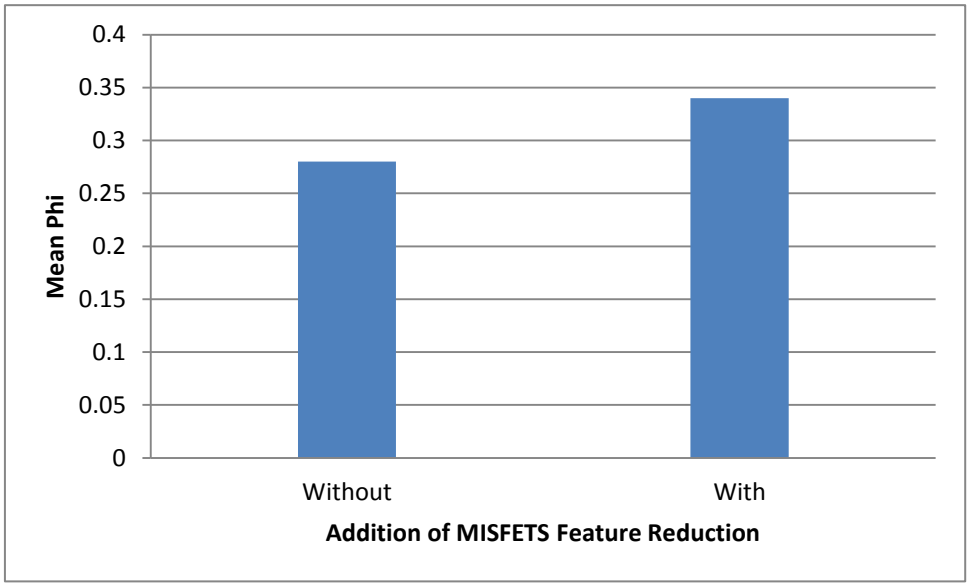

Figure 12.9: Results of SABIL features on single LDA classifier with 40 PCs with and without MISFETS for feature reduction

While ADEN and ADENZ demonstrated the advantage of supervised FS/R techniques on the Study A feature sets, SCRIS's best performance was still within the range of random guessing. The mean phi performance "bump" that SCRIS achieved by using the

mixed approach did not translate into an increase in performance by using MISFETS in conjunction with other methods.

The use of a single subject for training and the remainder for testing established the lower boundaries of performance for the LDA classifier. Even when using fewer subjects for training, the results were comparable to standard cross-validation. The potential limitations of subject-based training and testing on a linear classifier were demonstrated, as adding more subjects did not increase the mean phi correlation. However, the selection of feature indices from a single person using ADEN or MISFETS was enough to achieve a result comparable to the classifiers trained on a wider population. A microsleep detection system with even a rudimentary classifier may be able to select from among an arbitrary list of spectral bands and channels, yet still be able to function above a "random guess" phi value of 0.00 on individuals that it was not exposed to in training.

The approaches presented here demonstrated many similar limitations. The features selected in Study A were highly dependent upon differences in preprocessing, while the selected features in SCRIS were absent in between subjects. The shortcomings of MISFETS showed that while certain features corresponded to generalized differences, the feature selection alone was not enough to compensate for differences between subjects. The use of LDA also acted as a major limitation, as only FS/R techniques were compared. Changing the standard LOOCV evaluation approach also failed to demonstrate increases in performance, although a performance baseline was provided. However, MISFETS demonstrated that the total number of features could be reduced without losing key spectral information regarding brain-state, decreasing the computational resources required.

12.5 Summary

The specific ADEN and ADENZ indices calculated during the mixed cross-validation were compared with standard LOOCV. Additionally, RADEN, RADENZ, GADEN, and GADENZ were tested. The final results indicated that simply taking direct feature indices are insufficient for linear classifier training, although limiting the training size to one subject also demonstrated that training linear classifiers on larger numbers of individuals did not necessarily increase the performance. The results imply that a microsleep detection device preset to select arbitrary channels and features may perform at a rate above random guessing. While LDA has disadvantages and limitations, it can be highly robust under certain circumstances. Attempting to use LDA for microsleep prediction tested that robustness.

CHAPTER 13. PREDICTION OF MICROSLEEP EVENTS

13.1 Introduction

Previous research (Davidson et al., 2007; Peiris et al., 2011) with the Study A features focused exclusively on the detection of microsleeps during the event. However, theta band spectral changes in the EEG (Poudel et al., 2010) during the microsleeps were believed to offer a potential method for anticipating the onset of events. An EEG-based detector successfully able to anticipate microsleeps and detect occurring ones could save lives and reduce industrial accidents without the need for an eye-closure detecting camera.

According to the literature, spectral changes in the EEG occur up to 4 s prior to the microsleep (Poudel et al., 2010). In particular, changes occurred on the theta and alpha bands of EEG. Due to using a sliding window function, each 2-s segment contained information able to show spectral changes occurring at the onset of each microsleep. It was considered that spectral changes in the epochs corresponding to the pre-onset period could be sufficient to determine predict a microsleep.

Hypothesis 5: *Changes in the pre-onset period before microsleeps can be used to predict the onset of a microsleep.*

Rationale: Based upon the literature, changes in the theta and alpha bands occur up to 4 s before microsleeps (Davidson et al., 2007; Poudel et al., 2010). As a result, changes in spectral features could allow for EEG-based prediction of microsleeps.

13.2 Methods

A rudimentary method used to evaluate the effects of microsleep anticipation was simply to denote an arbitrary period of time before a microsleep as a microsleep event. Due to the binary nature of the "gold standard" based detectors utilized thus far, the change was simply converting observations preceding the interval between a microsleep from alert (0) to event (1). Initially, an onset of a single observation preceding a microsleep was used. This equates to preditction of the onset at the time of onset.

Based upon the literature, a pre-onset period of 1-s was believed to be sufficient (Poudel et al., 2010). While changes could occur up to 4 s prior to the microsleep, many occurred within 1-s prior to the microsleep. Since the spectral window was 2-s long with 50% overlap with the previous window, the result was the 1-s pre-onset was the centred before the

microsleep began. If a microsleep began at t=0, the pre-onset window would correspond to t=-1±1 s. If the prediction occurred at 2 s before onset, this equates to a prediction of 1 s.

Detection of the 1-s pre-onset period was also done without the remainder of expert rated microsleep event, to determine if the spectral features used were sufficient to capture the pre-onset of the microsleep. Mixing was not performed on the features so that results could be directly compared with published values using LOOCV, such as the phi of 0.39 achieved with a stacking ensemble (Peiris et al., 2011) and the phi of 0.38 achieved with an LSTM neural network (Davidson et al., 2007).

Five feature sets were employed: SABIL, SABIS, SARUS, SABUS, and SCRIS. Despite the artefact pruning of segments from the SABIS and SABIL features potentially creating gaps, it was included for completeness of comparison. As with other research, a binary model of brain-state was used (Davidson et al., 2007; Peiris et al., 2011). All were tested utilizing their previous gold standards, the existing gold standard with the 1-s pre-onset, and the 1-s pre-onset by itself (referred to as "predictive case"). During the predictive case, all existing events had their state changed to "0," as the classifier would only be trained on the pre-onset period. While the class imbalance was increased, the potential gains for success were immense. As depicted in Fig. 13.1, each is visually shown with respect to an existing event.

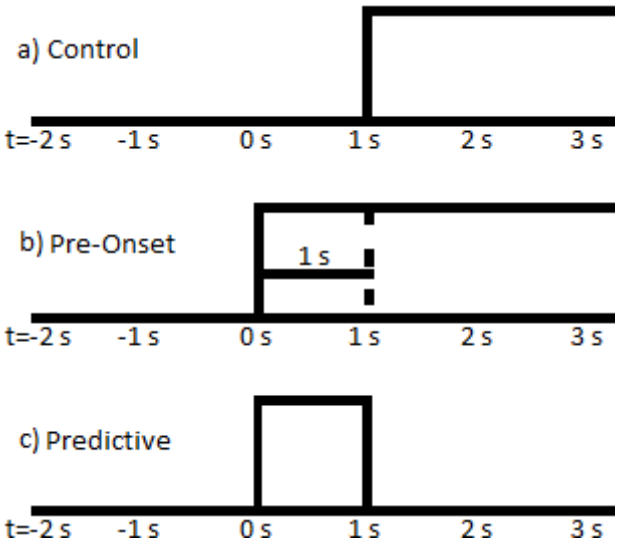

Figure 13.1: Schematic depiction of an event under 3 primary alternative gold standard scenarios including: a) Control, b) Pre-Onset, and c) Predictive

Single classifier LOOCV was used in concert with PCA, ADEN, ADENZ, and PLS. A total of 10 features or meta-features were retained initially, but the total number was gradually increased by increments of 10 to 100. A hypothesis was that the addition of the 1-s

pre-onset would not drastically alter the mean phi value, but the drastic reduction of "microsleep" events for the predictive case would make successful classification difficult due to further exacerbating the class imbalance. However, successful findings for the predictive case would suggest that existing spectral features were sufficient for successful classification of microsleeps.

13.3 Results

Comparison of standard LOOCV to the pre-onset data revealed no significant differences in performance. The addition of pre-onsets to the gold standard resulted in incrementally higher mean phi values in the case of all feature sets. The largest increase witnessed was the average increase from -0.02 (-0.11-0.12) to 0.03 (-0.07-0.12) for SCRIS with 10 features using ADENZ.

For Study A, the largest average in mean phi was a jump of 0.04 witnessed with 10 features were in the SABIS features with PLS and the SARUS features with ADENZ. The highest mean phi value from SCRIS with the pre-onset case was 0.09 (-0.02-0.23) with 20 ADEN features.

The highest mean phi value in all cases did not exceed 0.33 (0.12-0.52) with ADENZ with the SABIL features. The highest mean phi value for the SABUS features was 0.30 (0.03-0.53) by using ADEN with 90 features. The highest mean phi value for the SABIS features was 0.26 (-0.01-0.54) with 10 ADEN features. The highest mean phi for the SARUS features was 0.27 (0.06-0.41) with 30 ADEN features.

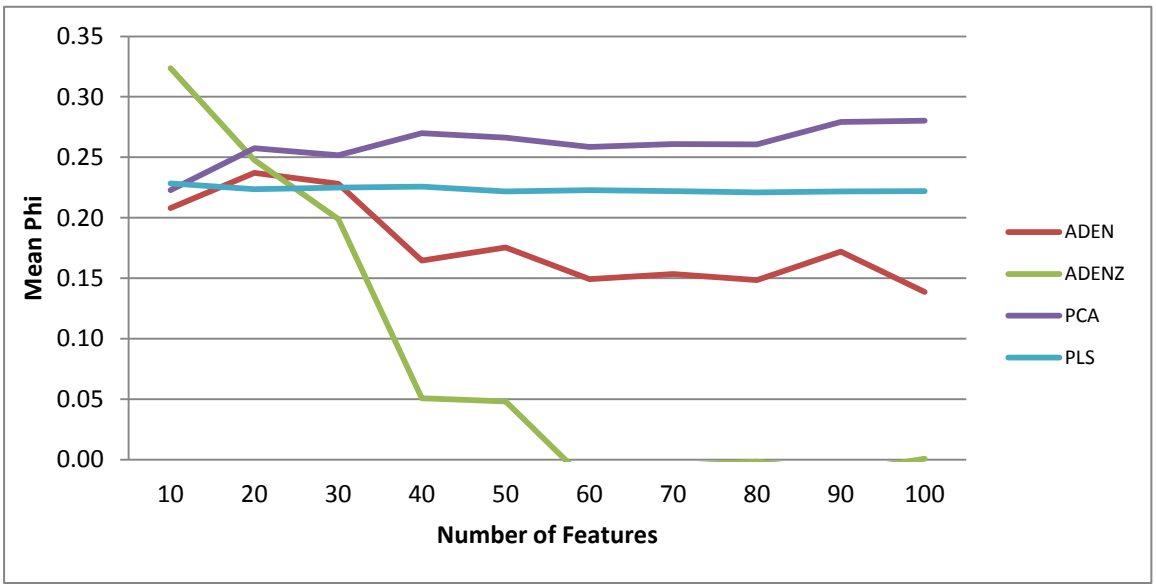

Figure 13.2: Results of SABIL features on a single LDA classifier with pre-onset data

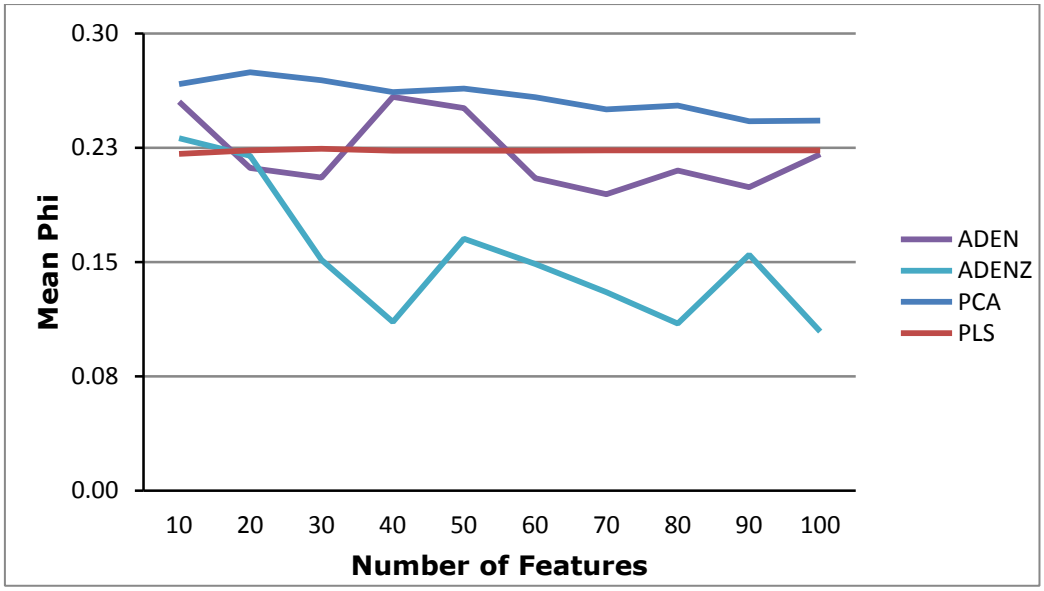

Figure 13.3: Results of SABIS features on a single LDA classifier with pre-onset data

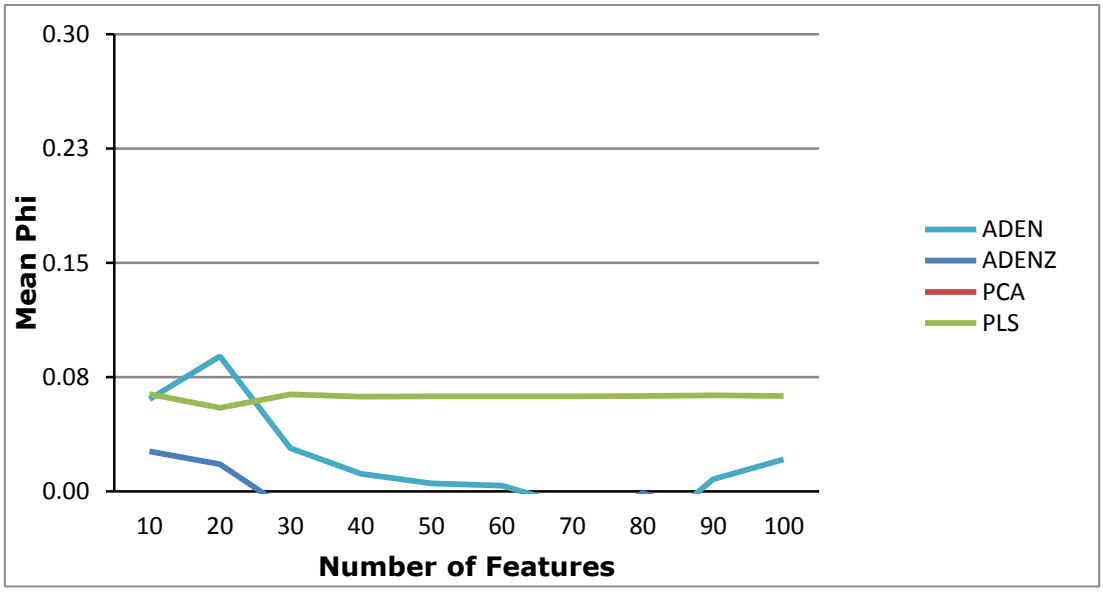

Figure 13.4: Results of SCRIS features on a single LDA classifier with pre-onset data

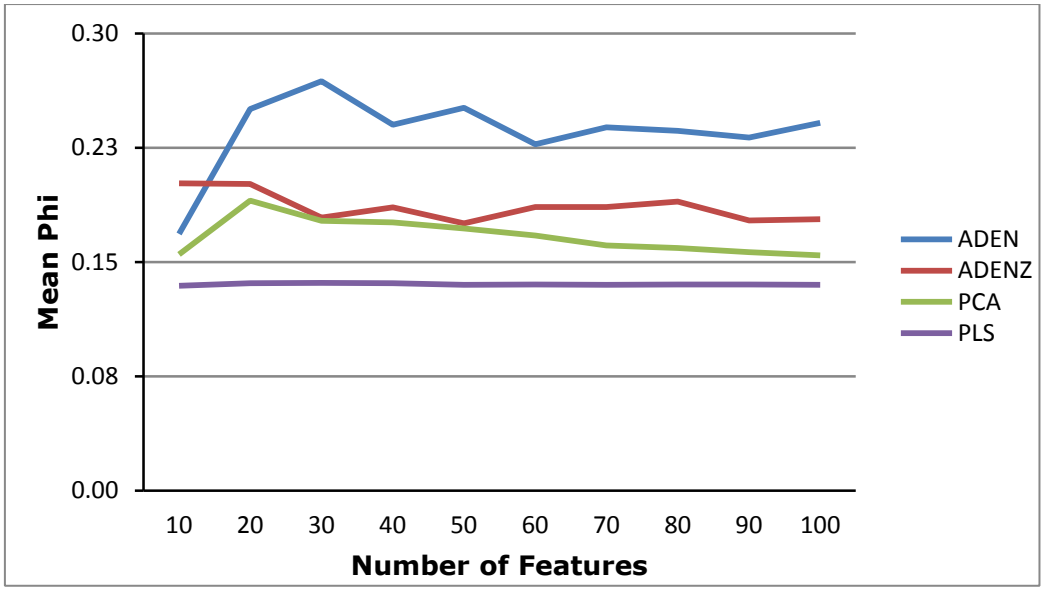

Figure 13.5: Results of SARUS features on a single LDA classifier with pre-onset data

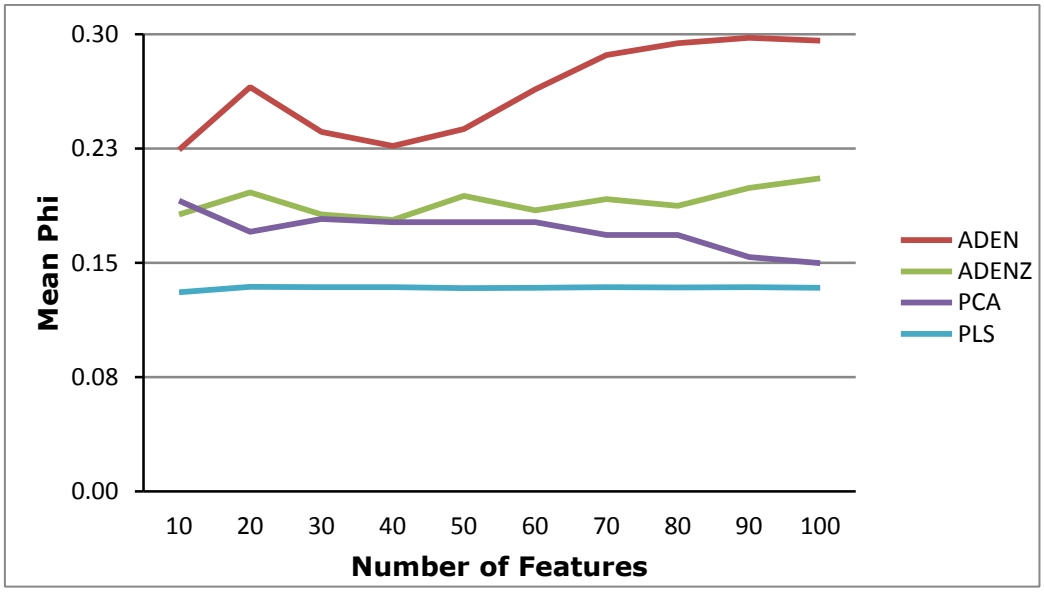

Figure 13.6: Results of SABUS features on a single LDA classifier with pre-onset data

The mean phi values dropped for Study A with the predictive case. The highest mean phi for the SABIS feature set corresponded to 0.05 with 10 ADEN features (-0.01-0.14). The highest mean phi for the SABUS feature set was PCA with 0.05 (-0.01-0.13) and 20 features. The highest mean phi for the SARUS features was 10 PCA features with 0.05 (-0.01-0.14). Once the predictive case data was investigated with the SCRIS features, the highest mean phi value dropped to 0.02 (-0.02-0.05) with 10 ADEN features.

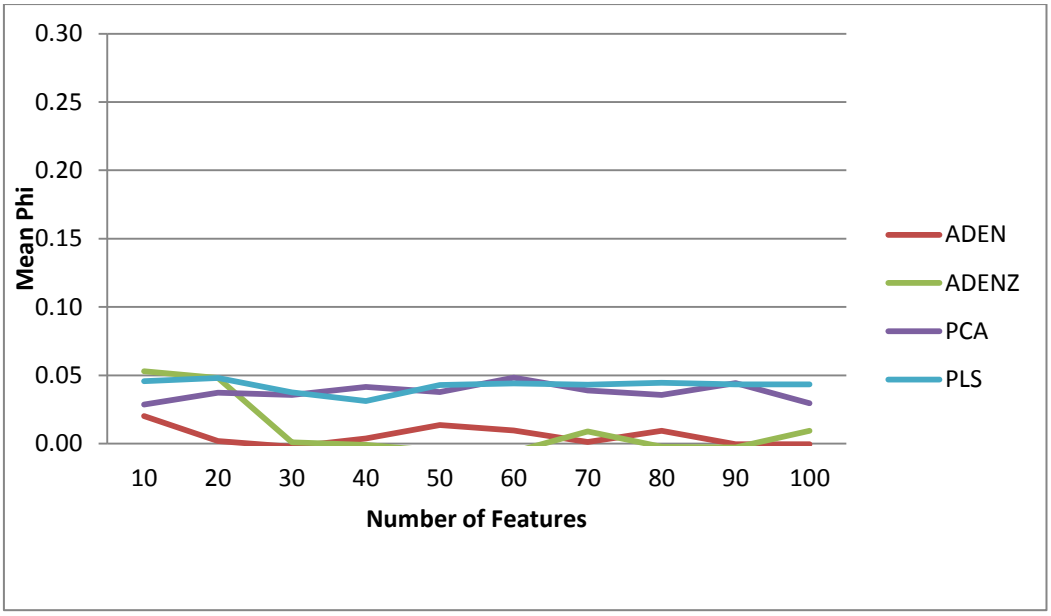

Figure 13.7: Results of SABIS features on a single LDA classifier with predictive data

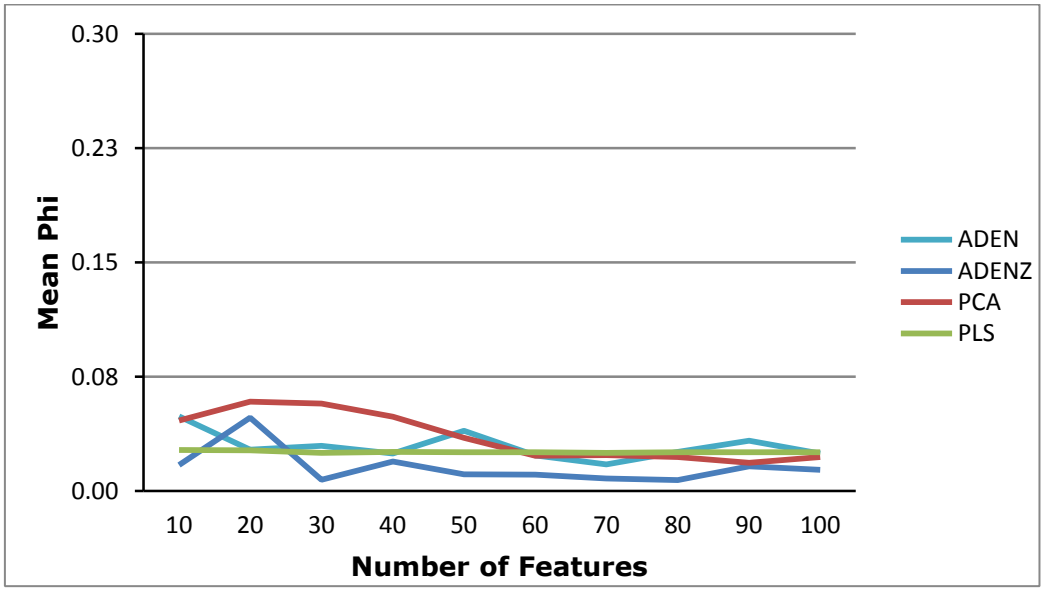

Figure 13.8: Results of SABIS features on a single LDA classifier with predictive data

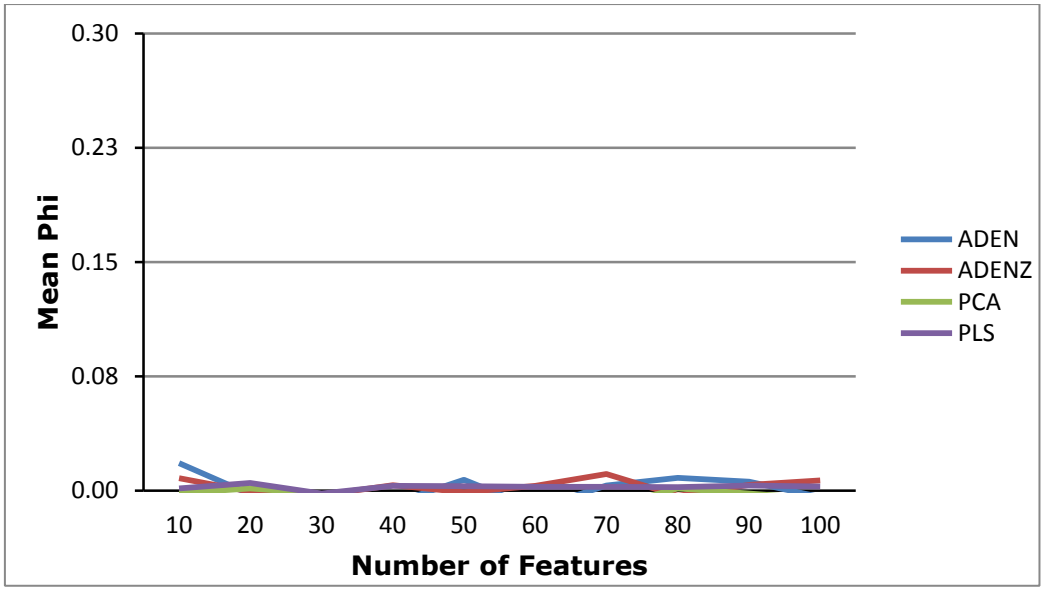

Figure 13.9: Results of SCRIS features on a single LDA classifier with predictive data

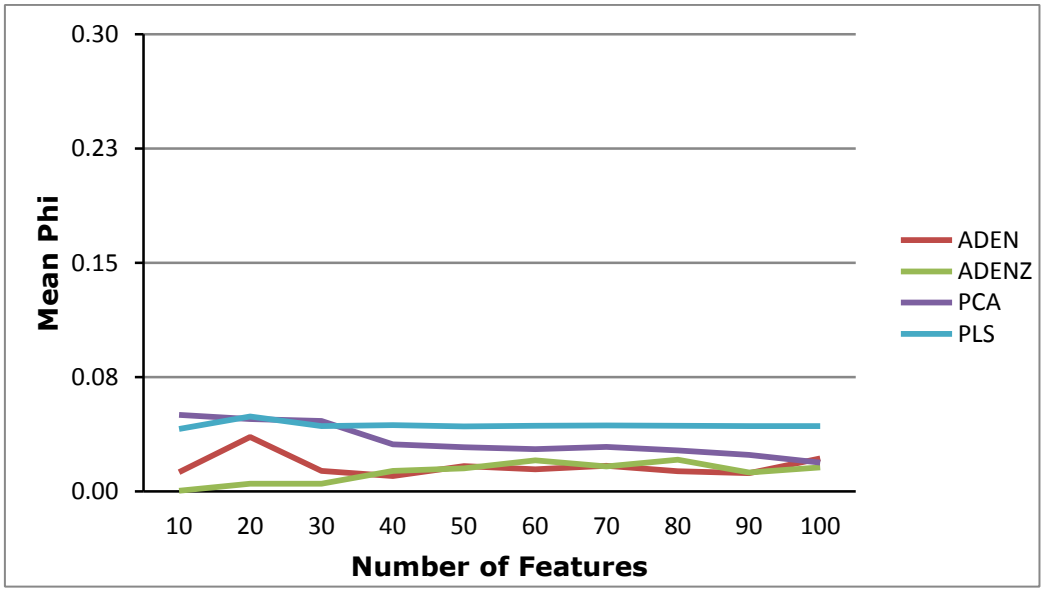

Figure 13.10: Results of SARUS features on a single LDA classifier with predictive data

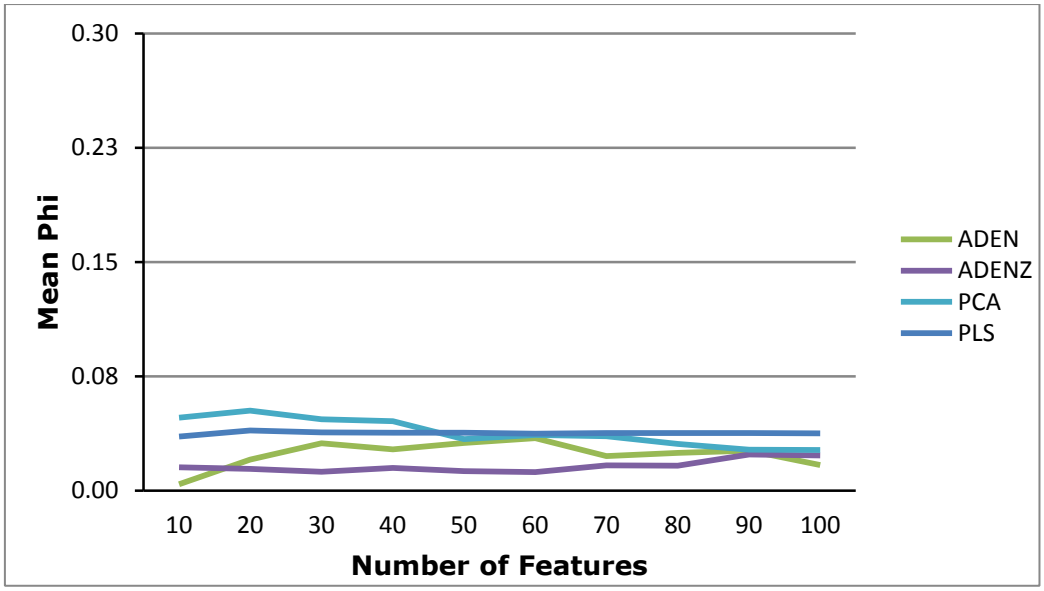

Figure 13.11: Results of SABUS features on a single LDA classifier with predictive data

Performance values universally decreased in cases involving prediction. The highest mean phi value from Study A corresponded to 0.05 from PCA on the bipolar and referential features.

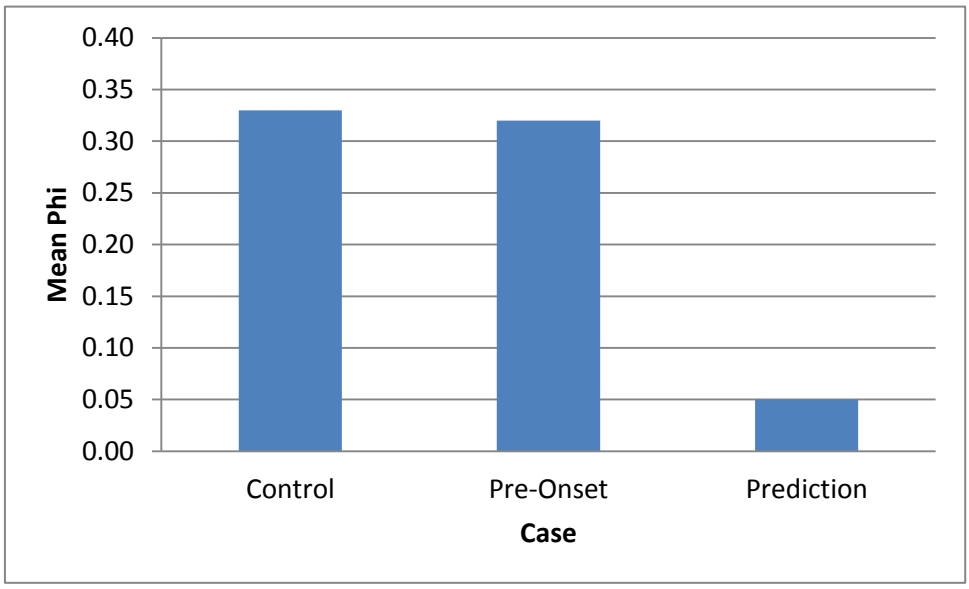

Figure 13.12: Results of SABIL features on a single LDA classifier with 10 ADENZ features

For Study C, the highest mean phi value was 0.02 with ADEN. Both values were lower than the control and pre-onset cases.

13.4 Discussion

The prediction of microsleeps remains a challenging task. Initial comparison between the control and pre-onset cases yielded no differences. There were trivial improvements, but these were not significant. While slight increases in performance were found, drawing

conclusions remained difficult. Conversely, the performance of each system configuration did not drop. A potential implication of this is that the arbitrary period before each event may be extended or adjusted to find the point at which the pre-onset periods begin to degrade performance. A closer examination of spectral features during the pre-onset period may potentially yield useful features under other circumstances, although the 1-s predictive case reduced the classifier to randomly guessing. Longer pre-onset periods were not used due to the failure of even 1-s pre-onset periods to gain meaningful results, as well as the absence of compelling evidence in the literature (Poudel et al., 2010). Additionally, the results demonstrated that correct classification can still occur with no drop in performance despite being trained on the pre-onset period events, signaling potential robustness of a real-time detector based on a simple linear classifier.

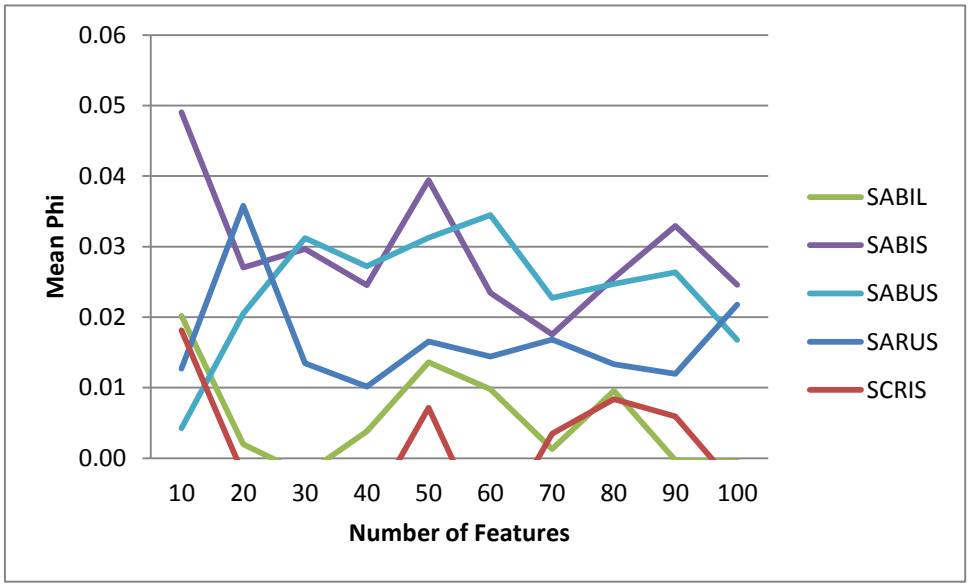

Figure 13.13: Comparative performance of ADEN with a single LDA classifier on major feature sets with prediction case

The failure of all system configurations to predict microsleeps potentially underscores the inadequacy of a single linear classifier. Many microsleep events had a duration corresponding to several observations due to lasting longer than 2 s. The predictive case reduced the presence of all events to a single observation. [CTHUHLU] Any discernible features within the microsleep brain-state might have been lost, and the dramatic imbalance between classes was further exacerbated. The presence of longer microsleep events reduced to a single observation that may lack the discernible features present within microsleep events, further limiting the effectiveness of a linear classifier.

The presented approaches to detecting microsleep pre-onset did not offer any improvement over prior systems. However, the inclusion of a brief pre-onset period did not

adversely affect the investigated system configurations. Due to the limitations of a state-based binary detection system with a single LDA classifier, the predictive approach would require further work. Potential avenues for improvement include the adaptation of an event detector rather than a state detector, use of a non-linear classifier, and additional preprocessing steps to improve the SNR. Alternatively, investigation of an eye-blink detector, perhaps integrating EOG, could be pursued.

13.5 Summary

Detecting the pre-onset of microsleeps is a key problem requiring continued investigation. The proposed method of adding an arbitrary 1-s pre-onset prior to the beginning of a microsleep did not adversely affect results, and may be attributable to trivial increases in mean phi performance. Performance aside, the pre-onset results also demonstrated that the classifier can be trained on a number of artificial "onset" events and still suffer no drop in performance. The 1-s "predictive" scenario, with the exclusion of the microsleep event, resulted in drastic drops in performance to random guessing in all cases. The use of a single LDA classifier as a binary state detector may be a limitation in successfully predicting the pre-onset of microsleep events. As such, future approaches may include the implementation of an event detector and potentially additional preprocessing steps.

CHAPTER 14. CONCLUSIONS

14.1 Key Findings

This thesis has presented novel methods of research, findings, and concepts, as well as providing the basis for future work:

- The results were highly dependent on feature extraction method, with log spectral features performing higher than linear spectral features.
- The use of ICA preprocessing and artefact pruning marginally improved detection performance, increasing phi from 0.35 to 0.40.
- Longer mean microsleep duration lead to higher mean within-subject phi values.
- The stacking ensemble corresponded to the highest performance on Study A, with a phi value of 0.40.
- A single LDA classifier achieved the highest performance on Study C, with a phi value of 0.10.
- The spectral features from the 1-s pre-onset period were insufficient for microsleep prediction.
- Spectral activity potentially relating to microsleeps occurred on the theta, alpha, beta, and gamma bands of EEG.
- Training on balanced data did not improve performance on microsleep detection.
- Much of the range in performance can be attributed to intra-subject differences.
- A new supervised feature selection method, ADEN, was proposed and refined into four primary separate variants ─ ADEN, ADENZ, GADEN, and GADENZ ─ which facilitated the search for spectral bands of note for analysis. ADEN often performed equivalent to or slightly higher than unsupervised PCA.

14.2 Review of Goals

The direction of research developed from the initial goals (Section 1.4). Of the initial goals, increasing the accuracy, sensitivity, and specificity of the detector above previous values in the literature (Davidson et al., 2007; Peiris et al., 2011) occurred only under circumstances outside of standard LOOCV, such as during the deployment of the "mixed subjects" cases. The reduction of system latency, however, rapidly veered into the improvement of feature selection/reduction methodologies. One of the proposed methods, ADEN, facilitated the detection of optimal channels and spectral bands for microsleep detection. Attempts at predicting microsleeps via an arbitrary pre-onset period revealed that a

system could retain appreciable performance even with a high number of artificially introduced events.

Table 14.1: Summary of key mean phi values for each primary topic: a) comparison of FS/R methods on "SNR = 0.3" artificial data; b) Replication of prior benchmarks; c) Comparison preprocessing for variant Study A features; d) Examining the effects of balanced data; e) Effects of mixing data on Study C; and f) Microsleep prediction outcomes.

a) Comparison of FS/R Methods — Artificial Events

PCA10	GADEN10	PLS10	ADEN10	CSP10
0.03	0.96	-0.01	0.94	0.08

-"SNR = 0.3" features run on Single LDA Classifier with LOOCV.

b) Replication of Benchmarks — Study A

Single Classifier	Stacking Ensemble
0.30	0.40

-SABIL features run on PCA FS/R and LDA Classifier with LOOCV.

-Prior Phi Benchmarks: Single Classifier = 0.31, Ensemble = 0.39

c) Comparison of Preprocessing — Study A

SABIL	SABIS	SABUL
0.40	0.36	0.35

-Features run on PCA FS/R and LDA Stacking Ensemble with LOOCV.

d) Comparing Balanced Data — Study A

Unbalanced	Balanced	Train on Balanced, Test on Unbalanced
0.33	0.52	0.33

-SABIL features run on ADENZ10 FS/R and Single LDA Classifier with LOOCV.

e) LOOCV and Mixed Data — Study C

LOOCV	Mixed
0.10	0.71

-SCRIS features run on ADEN FS/R and Single LDA Classifier with Mixing.

f) Microsleep Prediction — Study A

Standard Classifier	Pre-Onset	Prediction
0.33	0.32	0.05

-SABIL features run on ADENZ10 FS/R and Single LDA Classifier.

The main findings for each section are summarized in Table 14.1. ADEN and GADEN clearly outperformed all data on the artificial event data, as shown in (a). The stacking ensemble outperformed the single classifier, and other ensembles, when replicating the

benchmark performance, as shown in (b). The SABIL features outperformed the other feature sets, as shown in (c). Training on balanced data and testing on unbalanced data did not improve performance, as shown in (d). Mixing features was the only way the SCRIS features gained a high mean phi correlation, as shown in (e). Spectral features alone could not predict microsleep onset by training only on a pre-onset period, as shown in (f). However, the primary goal of much of the research was simply to increase performance.

14.2.1 Performance Improvement

The reported mean phi values of prior literature, 0.39 (Peiris et al., 2011) and 0.38 (Davidson et al., 2007), were achieved through either by complex classifiers (Davidson et al., 2007) and pruning of high-artefact data (Peiris et al., 2011). While single classifier performance was replicated with the SABIL features, the performance of the stacking ensemble on features without preprocessing was lower, even with the same feature extraction method. Variations in preprocessing accounted for a range of preprocessing values, but none surpassed the 0.40 (0.13-0.66) mean phi value reported with the SABIL features and the stacking ensemble using standard LOOCV.

The results achieved with the SABIL features were attributed to a combination of ICA preprocessing and artefact pruning. However, when variables involving feature extraction and preprocessing were changed, the performance of the stacking ensemble approached that of a single classifier. The stacking ensemble was able to perform higher on a standard LOOCV case, although ADEN and ADENZ performed higher when allowed to generalize using the mixed case even using a single classifier. ADENZ with 10 features achieved even slightly higher performance on a single classifier with the SABIL features, at 0.33 (0.12-0.52).

Performance over the 0.40 (0.13-0.66) mean phi value on Study A was achieved only by changing the terms of classification, such as mixing-based LOOCV or excluding "unpredictable" subjects. Mean phi values corresponding to successful classification of microsleep events on Study C was achieved only through similar efforts. The reasons for why the new techniques did not improve performance were theorized.

Variants in the preprocessing and feature extraction method used to generate features appeared to have a direct quantifiable effect. The 0.39 mean phi value reported by Peiris et al. (2011) depended on artefact pruning and ICA preprocessing in the training and testing, while the 0.38 reported by Davidson et al. (2007) demonstrate similar performance achievable without need for ICA. The techniques investigated highlight the necessity of improving the

feature extraction process for successful microsleep classification. The use of log power demonstrated an upward trend compared to features calculated with standard spectral power.

The use of PCA, ADEN, and ADENZ were often close to each other in terms of mean phi metrics when moderately successful classification was occurring. While ADEN and ADENZ are not necessarily orthogonal, they demonstrated that collinear features can increase robustness to achieve comparable performance, and even surpass PCA on the mixed cases. GADEN and GADENZ were still limited by the same problems as ADEN and ADENZ, so the additional layers of complexity did not provide the expected improvements. The use of ADEN and ADENZ as a bottleneck for GA and as a preprocessing method in MISFETS demonstrated the concepts perform at least equivalently to PCA and no worse. A key limitation with PLS is overfitting of linear models to the training data. While orthogonal like PCA, PLS lacks the robustness that simpler methods like PCA and ADEN can provide.

With single classifiers and ensembles, the ability of even a single LDA classifier to deliver the highest performance on feature sets lacking ICA preprocessing and artefact pruning was demonstrated, such as the SABUS, SARUS, and SARUL features. The use of classifier ensembles improved the other feature sets, except for the SCRIS features. Amongst the classifiers, the stacking ensemble was the highest performing.

Aside from LDA, the other pattern recognition algorithms did not improve performance. The RBF and SVM Gaussian kernel did not achieve the hoped for performance due to suboptimal clustering of datapoints. The SVM polynominal kernel is likely to have overfitted the training data (Omary, 2009).

Unlike bagging and boosting, the stacking ensemble could directly raise or lower weights to rank the performance of classifiers. The potential advantages of AdaBoost seemed to have been negated due to the large volume of incorrectly classified points that would accumulate and not provide a reliable classification boundary for microsleep data (Omary, 2009).

The microsleep detection problem is a difficult one due, at least in part, to the high variance between subjects. None of the techniques explored improved detection performance, but failed to surpass the phi correlation of 0.40 using the investigated methods. However, techniques like ADEN have potential value to increase speed of real-time execution by selecting the most relevant spectral features necessary for successful classification.

14.2.2 System Latency Reduction

Reduction of system latency was achieved by focusing on optimizing a smaller number of features. Supervised FS/R techniques were initially thought to result in superior performance to unsupervised FS/R techniques. ADEN and its variants initially performed well on the artificial feature sets, furthering this hypothesis. However, upon the EEG-derived spectral feature sets, differences in performance between supervised and unsupervised FS/R techniques were often negligible. Notwithstanding, a reduced number of features (< 200) would often correspond to the optimal mean phi performance in both cases before decreasing. As such, a small number of features can be used to ensure rapid classification by improving speed of execution. Even systems relying upon highly complex FS/R methods and classifiers (e.g., GADEN and SVM), operate quickly upon the conclusion of training.

The optimal classifier would have low latency, high sensitivity, and high selectivity. A simple LDA classifier and ensembles derived from it were employed in this research. The stacking ensemble (Peiris et al., 2011) and neural network (Davidson et al., 2007) used in prior cases required training on multiple subjects. As such, the actual classification process would be reduced to a binary decision if one was implemented in near real-time. The selection of particular generalized features, as indicated by the mixed feature data, was able to boost performances far higher than more complex classifiers. The mixed data case presented an alternative evaluation method to LOOCV, demonstrating rapid and higher accuracy classification when trained on random subsets of features from all subjects.

14.2.3 Optimizing Spatial and Spectral Information

Supervised feature selection methods like ADEN and ADENZ were able to identify specific electrodes, spectral bands, and other features of note. The most noteworthy specific features and channels varied greatly upon the feature set. The four separate Study A feature sets registered different channels, while Study C had innate limitations due to the relatively small number of common channels. The differences highlighted the innate issues with preprocessing EEG using different methods, and how even slight changes can drastically affect final performance.

The specific channels identified by FS/R were analysed for commonalities in spatial location. In addition to common channels, a search was undertaken for specific spectral bands continually selected for the Study A and Study C feature sets. Correlations between both

were highly sought for comparison with prior literature. Changes in the theta and alpha bands were of particular note, as were their respective power ratios (Poudel et al., 2010).

With the SABIL feature set, the top features calculated using ADEN and ADENZ included both the non-normalized and normalized alpha band spectral power from the bipolar channels T4-T6, P3-O1, and F3-C3. For individual subjects, the broader range of top features included gamma band power and normalized beta power. The optimal features for each individual included at least one of the top features for the group.

These top EEG features could provide more insight into brain-states. While spectral changes on the theta and alpha bands were known, the other changes were not (Davidson et al., 2007; Poudel et al., 2010). As a result of this, there is potential evidence of microsleep-related activity on the beta and gamma bands. However, the limitations of the SABIL features prevented further investigated.

In contrast to spectral bands, the specific electrodes selected was highly dependent upon the feature set. In Study A, variations in methods such as the use of ICA, artefact pruning, or use of referential or bipolar inputs could drastically change the results. Simultaneously, the divergences in performance between the referential and bipolar features were often close. In some instances, even the SARUS and SABUS feature sets had higher mean phis than the SABIS feature set, which had artefact pruning and ICA preprocessing performed on it. Surprisingly, as ADEN, ADENZ, and PCA often corresponded to the highest mean phi values, both supervised and supervised FS/R techniques proved able to assist with microsleep detection.

14.2.4 Microsleep Prediction

Predicting microsleep events was recognized as a highly desirable feature for a microsleep detection system. Prior literature suggested a method for prediction (Peiris et al., 2011), but a simpler and independent method was used to investigate the feasibility of microsleep detection. If linearly discernible EEG spectral features were present in an arbitrary period before the microsleep, it was hypothesized that preliminary detection of microsleep events by EEG alone was possible. The findings indicated that this was not the case. The 1-s pre-onset period was used due to the sliding window rating system, and during this interval, no EEG features were discernible for any of the feature sets. However, the microsleep detection software can be affected by false positives and continue to function.

While EEG alone may not have linearly discernible features, a combined input of multimodal signals might. As microsleeps consist of eye closure and drowsy behaviour (such as head nodding) (Lal and Craig, 2001; Peiris et al., 2006b; Golz and Sommer, 2010), two potential alternative vectors are computer vision-based eye closure detection with a camera and head movement detection using accelerometers. A weighted sum of features could determine the most reliable combination of features for an individual session, with an adaptive online classifier. The multimodal approach would require more features, but potentially combines the innate strengths of each signal type. However, such research is beyond the scope of this thesis.

14.3 Review of Hypotheses

Several findings over the course of this thesis are novel. The investigation presented a number of approaches over the course of the implementation of ICTOMI, expansion of Study A, and investigation of Study C.

Hypothesis 1: Simulated EEG events with a variable SNR provide an estimate of performance on real EEG feature sets.

Outcome: Hypothesis rejected. The performance of system configurations on simulated EEG events was not an accurate estimate of performance on real EEG feature data. The artificial event data with the 15-Hz sine pulse was intentionally synthetic when compared with the complexities of microsleep data, so that the spectral content and class balance could be completely controlled for software validation purposes. The chief evidence against this hypothesis speculated that based on the performance, a simple, supervised method like ADEN could be the best of the FS/R methods in terms of the mean phi value on LOOCV. The dramatic increase in performance corresponding to the use of ADEN was limited to the artificial event data. PLS occasionally would overfit to the training data, but ADEN and its variants achieved a rough approximation in mean phi values to PCA in most cases. The ability for unsupervised FS/R techniques to compete with a supervised system was demonstrated on each subject-based LOOCV.

Hypothesis 2: Artificially altering class balance for training will result in an increase in performance due to removing classifier bias from class imbalance.

Outcome: Hypothesis rejected. Training on balanced and testing on unbalanced did not significantly increase the performance of the investigated system configurations. While

the artificially balanced data increased the mean phi value, the increase was wholly artificial and resulted from the fact that the entire problem had been altered substantially. Due to the highly imbalanced nature of the real feature set, classifier performance still depended upon the specific subject being tested upon. As a result, the mean phi values of classifiers trained on unbalanced data ended up performing no better and no worse than training on unbalanced data. The testing subjects ended up displayed a large variance independent of the training methodology.

Hypothesis 3: There is a positive correlation between classifier performance and mean microsleep duration.

Outcome: Hypothesis proven. The hypothesis was confirmed in several ways. Differences in preprocessing resulted in different performance values for Study A, while strong evidence was found for correlation between mean microsleep duration and mean WS phi for both Study A and Study C. In the case of Study A, the SABIS and SABIL features unsurprisingly dominated the SARUS and SABUS in terms of mean phi values. While individual exceptions existed, the trend across the four feature sets has the SABIS and SABIL features typically with the highest mean phi values, the SABUS in second, and the SARUS values at the lowest. SCRIS performed lower than all of the Study A feature sets. Study A had a constant number of channels for all subjects, unlike Study C. Another factor of note was that spectral features from half the subjects in Study C could not be used to train a classifier to successfully test the other half, while Study A only had two such individuals. These mean "within-subject" phi values for undetectable subjects were correlated with having mean microsleep durations less than 1.5 s for Study A and 3 s for Study C. Even the number of microsleeps is less important than their average duration. A potential implication was that longer epoch durations allow more time for a classifier to detect any meaningful changes in spectral activity for a potential person.

Hypothesis 4: Selection of an optimal set of spectral features via MISFETS will boost microsleep detection performance.

Outcome: Hypothesis rejected. Use of MISFETS to reduce the Study A feature sets and Study C in size did not meaningfully increase mean phi performance, even in conjunction with other methods. MISFETS was used in different ways with the aim of improving classifier performance upon all five feature sets, but none of them succeeded in generating significantly improved results. Static feature indices, and thus specific channels or spectral

bands, alone were insufficient to boost average performance at the microsleep identification task. As a result, the hypothesis was rejected due to the innate issues with the training of classifiers.

Hypothesis 5: Changes in the pre-onset period before microsleeps can be used to predict the onset of a microsleep.

Outcome: Hypothesis rejected. The addition of an artificial, arbitrary 1-s pre-onset period did not significantly boost results of conventional microsleep events, and was insufficient on its own to result in any outcome above random guessing. While changes in the theta and alpha bands were expected (Poudel et al., 2010), any changes witnessed in the arbitrary 1-s pre-onset period were insufficient to provide meaningful results above random guessing in the five feature sets examined. Even simply reducing a feature set to a greatly reduced set of "useful" features did not result in significant improvements. However, the reduction of channels and spectral bands to particular ones of interest did not perform worse. The capacity to select an information-rich subset of specific channels and spectral bands could improve microsleep detector performance.

14.4 Critique

The research presented possessed a number of flaws and shortcomings. Aside from ADEN-derived methodologies, a key missing area was the relative lack of other novel techniques for classification and feature selection/reduction covered through the duration of the research. The novelty of the research was reduced due to the almost extensive reliance on LDA after other classifier algorithms were dismissed early on. Additionally, new types of FS/R, classifiers, and ensembles have been proposed in the literature as being more reliable and successful than linear systems. Many of these are biologically inspired, as well as including probability density functions, as compared to simple distance between two classes. As such, ICTOMI did not take full advantage of these in attempts to improve upon the base performance. A commercial product might require adaptive classifiers to gradually adjust to new users and separate sessions, which was a topic mentioned but not explored. Additionally, the simpler concept of a blink or EOG-based eye closure detection system was not thoroughly expounded on in the literature review or investigated concepts. Such a system might prove more robust than an EEG-based system.

The modular design of ICTOMI added further levels of complexity. The separation of FS/R and the classification step under the aim of modularity added more layers requiring

debugging than a handful of specialized configurations. A combined FS/R and classification step would solve both issues, but would require some changes to ICTOMI as described in the implementation chapter. Specialization might yield a more effective system, even at the cost of versatility.

The research largely overlooked a particular training technique better able to compensate for unbalanced datasets (Raudys, 1991; Omary, 2009): While artificially balanced data were investigated for training and testing, no algorithm proposed included a weighting scheme that prioritized microsleeps over alert periods. The stacking ensemble assigned weights based upon component classifiers rather than individual microsleep events. AdaBoost included a weighting system for individual observations, but did not surpass the stacking ensemble in performance.

Another deficiency was the reliance on mean phi correlation as the primary metric of performance. The phi correlation is similar to other widely-used measures (such as the kappa coefficient). As the microsleep detection problem is one with highly imbalanced class distributions, the implications of sensitivity and selectivity scores could also have been mentioned more frequently, especially in cases where two system configurations performed similar to each other.

Failing to meaningfully discern between the often close performances of system configurations additionally detracted from the thoroughness of the analysis. The few techniques that consistently outperform others had performance drops after changing feature sets. The performance variability of the feature sets prevented definitive conclusions on system configuration performance comparisons from being made. For example, if a given FS/R method exhibited drastic improvements with a single LDA classifier when compared with others, the same method would be combined with ensembles and non-linear classifiers to see the full range of improvements. While Study A feature sets exhibited changes in response to different preprocessing methods, no such changes were witnessed in Study C.

The innate issues with Study C and the SCRIS features resulting from the uneven number of channels could have additionally been handled in other ways than arbitrarily inserting zeros or an arbitrary number of non-zero values. While only 19 channels were common across all 10 subjects, their placement was clustered within a small portion of the head instead of covering the breadth of the scalp. While the use of “Not-a-Number” (NaN), interpolated channels, or arbitrary constants did not change the results, future work might list the mean phi from within-subjects classification alongside the conventional LOOCV values. Additionally, the exclusion of subjects from both studies for a lack of microsleeps was a

concern due to the loss of generalization on an already small experimental population. The finding of a correlation between lapse duration and within-subject phi could also have been further explored as a factor to improve detection performance.

The statistical distribution of results from the investigated system configurations rarely resulted in significant differences on the various EEG feature sets. Due to this, the implementation of code in ICTOMI was initially suspect. Additional validation on WEKA partially alleviated some of this concern, the failure of ensembles and the investigated supervised feature selection techniques to improve upon prior baseline performances remains troubling. Additionally, while ADEN would often select collinear features and channels, the use of more orthogonal techniques like GADEN or MISFETS did not produce improvements. The comparative effects of variant preprocessing techniques could have been additionally highlighted by the inconclusive performance values. Over-reliance on LDA-based classifiers might have also made over-fitting on the training data an issue. In summary, these issues presented the greatest potential shortcomings of the research.

14.5 Future Work

Despite the findings presented here, future work in the field of microsleep detection remains. Additional refinements on Study C may require a longer period of specialized research. Inclusion of specialized modules for ICTOMI directly integrating several simultaneous steps could potentially benefit future research.

Future focus on Study A could focus on the optimization of feature extraction and preprocessing techniques. While the SABIL features corresponded to the highest performance metrics, additional time could be spent on methods to specifically improve the performance of the "raw" (SARUS and SABUS) feature sets. The "raw" feature sets permitted an easier and direct method to observe any differences between bipolar and referential EEG without artefact pruning. The often close performances of the "raw" and "clean" (SABIL and SABIS) feature sets for Study A indicated that refinement of preprocessing may also be worth investigating. Further refinements of FS/R and classifier structures would preferably be carried out.

In many respects, Study C could be researched in far greater depth, and feature sets than SCRIS developed. Resolving the lack of constant channels between subjects could be improved through a systematic investigation of alternatives. For example, a constant number of the "most optimal" features could be selected for each subject as was done with MISFETS, perhaps drawn from within-subject analysis, and used for standard LOOCV analysis. Finally,

the potential ability of a classifier to detect microsleeps based upon mean duration should warrant further investigation.

Perhaps most importantly, other types of FS/R modules and classifiers should be investigated. A weighting system prioritizing the correct classification of microsleeps over alertness could also compensate for an imbalance between classes. Probabilistic systems and various weighting schemes could be investigated, as well as the direct integration of FS/R classification. For example, ICTOMI modules handling these (or more) aspects could be developed and compared alongside less-specialized counterparts. That way, the modularity of the system would not be compromised while expanding the range of options available. The use of adaptive classifiers and inclusion of multimodal signals is another step necessary for integration into a wearable, real-time microsleep detection system, as the device would need to adjust for each new user and session. In addition, completely novel systems, such as the biologically-inspired OpenWorm project (Palyanov, 2014) or a dynamic evolving neural-fuzzy interface system (DENFIS) model (Kasabov and Song, 2002), might be used for microsleep detection.

Additionally, microsleep events could be explored in relationship to spatio-temporal features. A feature matrix comprised of vectors, each consisting of 544 features for a 2-s window, potentially does not capture other relevant information involving the timing of microsleeps and relations between channels. As such, the type of spectral feature matrix used for much of this work may innately be limited in its ability to reflect the brain-state, potentially hindering progress in the field of microsleep detection. While wavelets did not improve epileptic spike detection performance (Goelz et al., 1999), a wealth of other spatio-temporal feature extraction techniques exist (Kasabov and Song, 2002; Chavez et al., 2003).

The possibility of microsleep related activity on the gamma and beta bands warrants further investigation. The limitations of the feature sets used include low number of subjects, a low number of sessions, and variable quality of EEG features for each individual. Despite this, a more thorough analysis of the higher frequency activity could provide new insights into brain-states. The use of spatio-temporal features may also improve provide insights about brain-states, including ways to personalize a microsleep detector.

A largely unexplored direction for microsleep research is the possibility of personalizing a detection system. Simple LDA classifiers achieved phi values up to 0.57 on one individual, which is substantially higher than the 0.40 from LOOCV. Another study may be necessary to further explore this direction, which would include a larger number of separate sessions per subject. Such data would provide a useful resource to further develop

personalized microsleep detectors. Personalized microsleep detectors are limited by the necessity of having an externally verified gold standard for recorded EEG. An automated gold standard consisting of multiple sensors may exist in the future, but such a system has not yet been investigated.

The research undertaken was more than the continuation of earlier work. Reconstruction of Study A from its raw components was undertaken. New elements included expanded research on Study A, the implementation of ICTOMI, and the analysis of Study C. Supervised FS/R techniques and complex classifiers were examined, often with some techniques taken directly from BCI. Future work remains for the challenge of microsleep detection, but inclusion of techniques from EEG-based BCI and machine learning should lead in a promising direction.

REFERENCES

Alba, N. A., Sclabassi, R. J., Mingui, S., & Cui, X. T. (2010). Novel Hydrogel-Based Preparation-Free EEG Electrode. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 18*(4), 415-423.

Alpaydin, E. (1992). Multiple neural networks and weighted voting. *Proceedings of 11th IAPR International Conference on Pattern Recognition Methodology and Systems, II*, 29-32.

Anderson, C., Wales, A. W. J., & Horne, J. A. (2010). PVT lapses differ according to eyes open, closed, or looking away. *Sleep, 33*(2), 197-204.

Bassani, T., & Nievola, J. C. (2008). Pattern recognition for brain-computer interface on disabled subjects using a wavelet transformation. *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2008*, 180-186.

Bergasa, L., Nuevo, J., Sotelo, M., Barea, R., & Lopez, M. (2006). Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems, 3*, 63-77.

Blankertz, B., Losch, F., Krauledat, M., Dornhege, G., Curio, G., & Muller, K. R. (2008). The Berlin brain-computer interface: accurate performance from first-session in BCI-naive subjects. *IEEE Transactions on Biomedical Engineering, 55*(10), 2452-2462.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

Chavez, M., Le Van Quyen, M., Navarro, V., Baulac, M., & Martinerie, J. (2003). Spatio-temporal dynamics prior to neocortical seizures: amplitude versus phase couplings. *IEEE Transactions on Biomedical Engineering*, 50(5), 571-583.

Chee, M. W., Tan, J. C., Zheng, H., Parimal, S., Weissman, D. H., Zagorodnov, V., & Dinges, D. F. (2008). Lapsing during sleep deprivation is associated with distributed changes in brain activation. *Journal of Neuroscience, 28*(21), 5519-5528.

Chen, Q. K., U. (2005). Analysis of extended partial least squares for monitoring large-scale processes. *IEEE Transactions on Control Systems Technology, 13*(5), 807-813

Chen, X., He, C., Wang, Z.J., & McKeown, M.J. (2013). An IC-PLS framework for group corticomuscular coupling analysis. *IEEE Transactions on Biomedical Engineering, 60*(7), 2022-2033.

Chundi, G. S., Lloyd-Hart, M., & Sundareshan, M. K. (2004). Training multilayer perceptron and radial basis function neural networks for wavefront sensing and restoration of turbulence-degraded imagery. *Proceedings of the IEEE International Joint Conference on Neural Networks, 3*, 2117-2122.

Conradt, R., Brandenburg, U., Penzel, T., Hasan, J., Varri, A., & Peter, J. H. (1999a). Vigilance transitions in reaction time test: a method of describing the state of alertness more objectively. *Clinical Neurophysiology, 110*(9), 1499-1509.

Conradt, R., Brandenburg, U., Penzel, T., J Hasan, J., Värri, A., & Peter, J. (1999b). Vigilance transitions in reaction time test: a method of describing the state of alertness more objectively. *Clinical Neurophysiology, 110*(9), 1499-1509.

Coufal, D. (2009). Redundant rules detection in EEG fuzzy classifier. *Intelligent Engineering Systems, 2009*, 177-181.

Davidson, P., & Jones, R. (2007). EEG spectral dynamics for lapse detection [Abstract]. *Sleep and Biological Rhythms, 5*(Suppl 1), A39.

Davidson, P. R., Jones, R. D., & Peiris, M. T. R. (2005). Detecting behavioral microsleeps using EEG and LSTM recurrent neural networks. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 27*, 5754-5757.

Davidson, P. R., Jones, R. D., & Peiris, M. T. R. (2007). EEG-based lapse detection with high temporal resolution. *IEEE Transactions on Biomedical Engineering, 54*, 832-839

De Gennaro, L., Ferrara, M., Ferlazzo, F., & Bertini, M. (2000). Slow eye movements and EEG power spectra during wake-sleep transition. *Clinical Neurophysiology, 111*, 2107-2115.

Dehbaoui, A., Tiran, S., Maurine, P., Standaert, F., & Veyrat-Charvillon, N. (2011). Spectral Coherence Analysis - First Experimental Results. *IACR Cryptology ePrint Archive, 2011*, 56.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods, 134*, 9-21.

Deng, W., Qinghua, Z.,; Lian, S., Chen, L., & Wang, X. (2011). Projection vector machine: One-stage learning algorithm from high-dimension small-sample data. *Proceedings from the Annual 2010 International Joint Conference on Neural Networks, 2010*, 1-8.

Dinges, D. F., & Grace, R. (1998). PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. Technical Report (FHWA-MCRT-98-006). Washington, DC: US Dept. Transportation, Federal Highway Admin.

Dinges, D. F., & Powell, J. P. (1985). Microcomputer analysis of performance on a portable, simple visual RT task during sustained operation. *Behavior Research Methods, Instruments, & Computers, 17*, 652-655.

Dobrea, M., Dobrea, D. M., & Alexa, D. (2010). Spectral EEG features and tasks selection process: Some considerations toward BCI applications. *Proceedings of the IEEE International Workshop on Multimedia Signal Processing, 12*, 150-155.

Doran, S. M., Van Dongen, H. P., & Dinges, D. F. (2001). Sustained attention performance during sleep deprivation: evidence of state instability. *Archives Italiennes de Biologie, 139*(3), 253-267.

Dorrian, J., Rogers, N. L., & Dinges, D. F. (2005). Psychomotor vigilance performance: neurocognitive assay sensitive to sleep loss. In C. Khushida (Ed.), *Sleep Deprivation: Clinical Issues, Pharmacology and Sleep Loss Effects* (pp. 39-70). New York: Marcel Dekker Inc.

Duffy, F. H. (1989). Clinical value of topographic mapping and quantified neurophysiology. *Archives of Neurology, 46*, 1133-1134.

Faradji, F., Ward, R. K., & Birch, G. E. (2010). A simple approach to find the best wavelet basis in classification problems. *Proceedings of International Conference on Pattern Recognition, 20*, 641-644.

Finan, R. A., Sapeluk, A. T., & Damper, R. I. (1996). Comparison of multilayer and radial basis function neural networks for text-dependent speaker recognition. *IEEE International Conference on Neural Networks, 4*, 1992-1997.

Freund, Y., & Schapire, R. E. (1997). Decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences, 55*, 119-139.

Gandhi, H., Green, D., Kounios, J., Clark, C. M., & Polikar, R. (2006). Stacked generalization for early diagnosis of Alzheimer's disease. *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 28*, 5350-5353.

Gareis, I. E., Acevedo, R. C., Atum, Y. V., Gentiletti, G. G., Banuelos, V. M., & Rufiner, H. L. (2011). Determination of an optimal training strategy for a BCI classification task with LDA. *Proceedings of International IEEE/EMBS Conference on Neural Engineering, 5*, 286-289.

Geva, A. B. (1998). Feature extraction and state identification in biomedical signals using hierarchical fuzzy clustering. *Medical & Biological Engineering & Computing, 36*, 608-614.

Goelz, H., Jones, R. D., & Bones, P. J. (1999). Continuous wavelet transform for the detection and classification of epileptiform activity in the EEG. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 21*, 941.

Golz, M., Sommer, D., & Mandic, D. (2005). Microsleep detection in electrophysiological signals. *Proceedings of International Workshop on Biosignal Processing and Classification, 1*, 102-109.

Golz, M., & Sommer, D. (2010). Monitoring of drowsiness and microsleep. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 32*, 1787.

Golz, M., Sommer, D., Chen, M., Trutschel, U., & Mandic, D. (2007). Feature fusion for the detection of microsleep events. *Journal of VLSI Signal Processing, 49*(2), 329-342.

Golz, M., Sommer, D., Seyfarth, A., Trutschel, U., & Moore-Ede, M. (2001). Application of vector-based neural networks for the recognition of beginning microsleep episodes with an eyetracking system. *Proceedings of the Computational Intelligence: Methods and Applications, 2*, 1-5.

Greenwald, S. D., Smith, C. P., Sigl, J. C., Cai, H. M., & Devlin, P. H. (1999). The EEG Bispectral Index (TM): development and utility. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 21*, 443-443.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1).

Hanowski, R. J., Hickman, J., Fumero, M. C., Olson, R. L., & Dingus, T. A. (2007). The sleep of commercial vehicle drivers under the 2003 hours-of-service regulations. *Accident Analysis & Prevention, 39*(6), 1140-1145.

Harrison, Y., & Horne, J. A. (1996). Occurrence of 'microsleeps' during daytime sleep onset in normal subjects. *Electroencephalography and Clinical Neurophysiology, 98*(5), 411-416.

Hazewinkel, M. (2011). Greedy algorithm. *Encyclopedia of Mathematics.* Retrieved from http://www.encyclopediaofmath.org/index.php?title=Greedy_algorithm&oldid=11679

Hoeve, M. J., Jones, R. D., Carroll, G. J., & Goelz, H. (2001). Automated detection of epileptic seizures in the EEG. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 23*, 943-946

Honorio, J., Tomasi, D., Goldstein, R., Leung, H., & Samaras, D. (2012). Can a single brain region predict a disorder? *IEEE Transactions on Medical Imaging, PP*(99), 1-1.

Hori, T., Hayashi, M., & Morikawa, T. (1994). Topographical EEG changes and the hypnagogic experience. In R. D. Ogilvie & J. R. Harsh (Eds.), *Sleep Onset: Normal and Abnormal Processes* (pp. 237–253). Washington, DC: American Psychological Association.

Hutapea, D. K. Y., M.Z.; Asirvadam, V.S. (2014). Single trial visual evoked potential extraction using partial least squares-based approach. *IEEE Journal of Biomedical and Health Informatics*(99), 1.

Innes, C. R. H., Poudel, G. R., Signal, T. L., & Jones, R. D. (2010). Behavioural microsleeps in normally-rested people. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 32*, 4448-4451.

James, C. J., Hagan, M. T., Jones, R. D., Bones, P. J., & Carroll, G. J. (1997). Multireference adaptive noise cancelling applied to the EEG. *IEEE Transactions on Biomedical Engineering, 44*, 775-779.

James, C. J., Jones, R. D., Bones, P. J., & Carroll, G. J. (1999). Detection of epileptiform discharges in the EEG by a hybrid system comprising mimetic, self-organized artificial neural network, and fuzzy logic stages. *Clinical Neurophysiology, 110*, 2049-2063.

Jones, R. D., Poudel, G. R., Innes, C. R. H., Davidson, P. R., Peiris, M. T. R., Malla, A., Signal, T. L., Carroll, G. J., Watts, R., & Bones, P. J. (2010). Lapses of responsiveness: Characteristics, detection, and underlying mechanisms. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 32*, 1788-1791.

Jung, T.-P., Huang, K.-C., Chuang, C.-H., Chen, J.-A., Ko, L.-W., Chiu, T.-W., & Lin, C.-T. (2010). Arousing feedback rectifies lapse in performance and corresponding EEG power spectrum. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 32*, 1792-1795.

Jung, T.-P., Makeig, S., Stensmo, M., & Sejnowski, T. J. (1997). Estimating alertness from the EEG power spectrum. *IEEE Transactions on Biomedical Engineering, 44*(1), 60-69.

Lakany, H. & Conway, B.A. (2006). Classification of Wrist Movements using EEG-based Wavelets Features. *Proceedings of the Annual International IEEE-EMBS Conference of the Engineering in Medicine and Biology Society*, 27, 5404-5407.

Kasabov, N., & Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Transactions on Fuzzy Systems, 10(2), 144-154.

Kim, H. D., Park, C. H., Yang, H. C., & Sim, K. B. (2006). Genetic algorithm based feature selection method for pattern recognition. *Proceedings of SICE-ICASE International Joint Conference, 2006*, 1020-1025.

Kirk, B. P., & LaCourse, J. R. (1996). Detecting lapses in visual awareness from the EEG power spectrum with a neural network. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 22*, 117-118.

Kiymik, M., Akin, M., & Subasi, A. (2004). Automatic recognition of alertness level by using wavelet transform and artificial neural network. *Journal of Neuroscience Methods, 139*, 231-240.

Krajewski, J., Batliner, A., & Wieland, R. (2008). Multiple classifier applied on predicting microsleep from speech. *Proceedings of the International Conference on Pattern Recognition, 19*, 1-4.

Lal, S. K. L., & Craig, A. (2001). A critical review of the psychophysiology of driver fatigue. *Biological Psychology, 55*(3), 173-194.

Leong, W. Y., Mandic, D. P., Golz, M., & Sommer, D. (2007). Blind extraction of microsleep events. *Proceedings of the International Conference on Digital Signal Processing, 15*, 207-210.

Lijing, M., Jing, J., & Xingyu, W. (2012). A comparison of navigation system based on P300 BCI and SSVEP BCI. *Proceedings of Chinese Control and Decision Conference, 24*, 3703-3708.

Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2009). Regularized common spatial patterns with generic learning for EEG signal classification. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 31, 2009*.

Mackworth, N. H. (1957). Some factors affecting vigilance. *Advancements in Science, 53*, 389-393.

Makeig, S., & Inlow, M. (1993). Lapses in alertness: coherence of fluctuations in performance and EEG spectrum. *Electroencephalography and Clinical Neurophysiology, 86*(1), 23-35.

Malla, A. M., Davidson, P. R., Bones, P. J., Green, R., & Jones, R. D. (2010). Automated video-based measurement of eye closure for detecting behavioral microsleep. *Proceedings of International Conference of the IEEE Engineering in Medicine and Biology Society, 32*, 6741-6744.

Moser, D., Anderer, P., Gruber, G., Parapatics, S., Loretz, E., Boeck, M., Kloesch, G., Heller, E., Schmidt, A., Danker-Hopfe, H., Saletu, B., Zeitlhofer, J., & Dorffner, G. (2009). Sleep classification according to AASM and Rechtschaffen & Kales: Effects on sleep scoring parameters. *Sleep, 32*(2), 139–149.

Muradore, R. F., P. (2012). A PLS-based Statistical approach for fault detection and isolation of robotic manipulators. *IEEE Transactions on Industrial Electronics, 59*(8), 3167 - 3175.

Omary, Z. M., F. (2009). Dataset threshold for the performance estimators in supervised machine learning experiments. *Conference Proceedings from the Annual International Conference for Internet Technology and Secured Transactions, 2009*, 1-8.

Opitz, D., & Machin, R. (1999). Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research, 11*, 169-198.

Othman, M., Wahab, A., & Khosrowabadi, R. (2009). MFCC for robust emotion detection using EEG. *Proceedings of IEEE Malaysia International Conference on Communications, 9*, 98-101.

Palyanov, A., Balazs, S., Idili, G., Hokanson, J., Cantarelli, M., Currie, M., Gleeson, P., Khayrulin, S., & Larson, S. (2014). *OpenWorm*. Retrieved, 2014, from http://www.openworm.org/index.html

Parasuraman, R., Warm, J. S., & See, J. E. (1998). Brain systems of vigilance. In *The Attentive Brain* (pp. 221–256): MIT Press.

Park, Y., Luo, L., Parhi, K.K., and Netoff, T. (2011). Seizure prediction with spectral power of EEG using cost-sensitive support vector machines. *Epilepsia, 52*, 1761-1770.

Parini, S., Maggi, L., & Andreoni, G. (2007). An automated method for relevant frequency bands identification based on genetic algorithms and dedicated to the Motor Imagery BCI protocol. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 29*, 2512-2515.

Peiris, M. T., Davidson, P. R., Bones, P. J., & Jones, R. D. (2011). Detection of lapses in responsiveness from the EEG. *Journal of Neural Engineering, 8 (016003)*(1), 1-15.

Peiris, M. T. R., Jones, R. D., Davidson, P. R., & Bones, P. J. (2006a). Detecting behavioral microsleeps from EEG power spectra *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 28*, 5723-5726

Peiris, M. T. R., Jones, R. D., Davidson, P. R., & Bones, P. J. (2008). Event-based detection of lapses of responsiveness. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 30*, 4960-4963.

Peiris, M. T. R., Jones, R. D., Davidson, P. R., Bones, P. J., & Myall, D. J. (2005a). Fractal dimension of the EEG in detection of behavioural microsleeps. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 27*, 5742-5745.

Peiris, M. T. R., Jones, R. D., Davidson, P. R., Carroll, G. J., & Bones, P. J. (2006b). Frequent lapses of responsiveness during an extended visuomotor tracking task in non-sleep deprived subjects. *Journal of Sleep Research, 15*(3), 291-300.

Peiris, M. T. R., Jones, R. D., Davidson, P. R., Carroll, G. J., Signal, T. L., Parkin, P. J., van den Berg, M., & Bones, P. J. (2005b). Identification of vigilance lapses using EEG/EOG by expert human raters. *Proceedings of the Annual International Conference of IEEE Engineering in Medicine and Biology Society, 27*, 5735-5738.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*, 1226-1238.

Pomfrett, C. J. D., & Pearson, A. J. (1996). EEG monitoring using bispectral analysis. *Proceedings of IEEE Colloquium on New Measurements and Techniques in Intensive Care, 179*, 1-3.

Posner, M. I., & Petersen, S. E. (1990). The attention system of the human brain. *Annual Review of Neuroscience, 13*, 25-42.

Poudel, G. R., Innes, C. R. H., Bones, P. J., & Jones, R. D. (2010). The relationship between behavioural microsleeps, visuomotor performance and EEG theta. *Proceedings of Annual International Conference of IEEE Engineering in Medicine and Biology Society, 32*, 4452-4455.

Poudel, G. R., Innes, C. R. H., Bones, P. J., Watts, R., & Jones, R. D. (2014). Losing the struggle to stay awake: Divergent thalamic and cortical activity during microsleeps. *Human Brain Mapping, 35*, 257-269.

Poudel, G. R., Jones, R. D., & Innes, C. R. H. (2008). A 2-D pursuit tracking task for behavioural detection of lapses. *Australasian Physical and Engineering Sciences in Medicine, 31*(4), 528-529.

Qiao, X., Wang, Y., Li, D., & Tian, L. (2010). Feature extraction and classifier evaluation of EEG for imaginary hand movements. *Proceedings of the International Conference on Natural Computation, 2010, 6*, 2112-2116.

Quitadamo, L. R., Abbafati, M., Saggio, G., Cardarilli, G. C., Marciani, M. G., & Bianchi, L. (2009). Efficiency of a BCI system in a visual P300 protocol with different stimulation intervals. *Proceedings of the International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 1*, 670-673.

Raudys, S. J. J., A. K. (1991). Small sample size effects in statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(3), 252-264.

Reason, J. (1984). Lapses of attention in everyday life. In R. Parasuraman & D. R. Davies (Eds.), *Varieties of Attention* (pp. 515-549). Orlando: Academic Press.

Reshef, D., Reshef, Y., Finucane, H., Grossman, S., McVean, G., Turnbaugh, P., Lander, E., Mitzenmacher, M., & Sabeti, P. (2011). Detecting novel associations in large datasets. *Science, 334*, 1518-1524.

Robbins, T. W., James, M., Owen, A. M., Sahakian, B. J., Lawrence, A. D., McInnes, L., & Rabbitt, P. M. (1998). A study of performance on tests from the CANTAB battery sensitive to frontal lobe dysfunction in a large sample of normal volunteers: implications for theories of executive functioning and cognitive aging. Cambridge Neuropsychological Test Automated Battery. *Journal of the International Neuropsychological Society, 4*(5), 474-490.

Ruping, S. (2001). Incremental learning with support vector machines. *Proceedings of the IEEE International Conference on Data Mining, 2001, 1*, 641-642.

Sarter, M., Givens, B., & Bruno, J. P. (2001). The cognitive neuroscience of sustained attention: where top-down meets bottom-up. *Brain Research Review, 35*(2), 146-160.

Schapire, R. E., Rochery, M., Rahim, M., & Gupta, N. (2005). Boosting with prior knowledge for call classification. *IEEE Transactions on Speech and Audio Processing, 13*(2), 174-181.

Selim, A. E., Wahed, M. A., & Kadah, Y. M. (2009). Machine learning methodologies in P300 speller Brain-Computer Interface systems. *Proceedings of the National Radio Science Conference, 26*, 1-9.

Shoaie, Z., Esmaeeli, M., & Shouraki, S. B. (2006). Combination of multiple classifiers with fuzzy integral method for classifying the EEG signals in brain-computer interface.

Proceedings of the International Conference on Biomedical and Pharmaceutical Engineering, 3, 157-161.

Sommer, D., Golz, M., Trutschel, U., Ramsthaler, C., & Moore-Ede, M. (2001). Characterization of the electroencephalogram of microsleep using self-organized feature maps. *ICSC Conference Proceedings on Advanced Computing in Biomedicine, 1*.

Steriade, M. (1996). Arousal: revisiting the reticular activating system. *Science, 272*(5259), 225-226.

Suykens, J., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing, 48*, 85-105.

Székely, G., Rizzo, M., & Bakirov, N. (2007). Measuring and testing dependence by correlation of distances. *Annuals of Applied Statistics, 35*, 2769-2794.

Székely, G., & Rizzo, M. (2009). Brownian distance covariance. *Annuals of Applied Statistics, 3*, 1236-1265.

Sun, S.-L., Xu, J.-H., Yu, L.-Y., Chen, Y.-G., & Fang, A.-L. (2008). Mixtures of common spatial patterns for feature extraction of EEG signals. *Conference Proceedings on Machine Learning and Cybernetics, 5*.

Takenouchi, T., & Eguchi, S. (2004). Robustifying adaboost by adding the naïve error rate. *Neural Computation, 16*(4), 767-787.

Torsvall, L., & Akerstedt, T. (1987). Sleepiness on the job: continuously measured EEG changes in train drivers. *Electroencephalography and Clinical Neurophysiology, 66*(6), 502-511.

Valley, V., & Broughton, R. (1983). The physiological (EEG) nature of drowsiness and its relation to performance deficits in narcoleptics. *Electroencephalography and Clinical Neurophysiology, 55*(3), 243-251.

Van Orden, K., Jung, T., & Makeig, S. (1999). Combined eye activity measures accurately estimate changes in sustained visual task performance. *Biological Psychology, 52*, 221-240.

Vuckovic, A., Radivojevic, V., Chen, A., & Popovic, D. (2002). Automatic recognition of alertness and drowsiness from EEG by an artificial neural network. *Medical Engineering & Physics, 24*(5), 349-360.

Wang, L., Xu, G., Wang, J., Yang, S., & Yan, W. (2011). Motor imagery BCI research based on Hilbert-Huang Transform and Genetic Algorithm. *International Conference on Bioinformatics and Biomedical Engineering, 5*.

Ward, D. M., Jones, R. D., Bones, P. J., & Carroll, G. J. (1999). Enhancement of deep epileptiform activity in the EEG via 3-D adaptive spatial filtering. *IEEE Transactions on Biomedical Engineering, 46*(6), 707-716.

Williams, G. W. (1963). Highway hypnosis: an hypothesis. *International Journal of Clinical and Experimental Hypnosis, 11*, 143-151.

Wolpert, D. (1992). Stacked generalization. *Neural Networks, 5*, 241-259.

Xu, H., Song, W., Hu, Z., Chen, C., Zhao, X., & Zhang, J. (2010). A speedup SVM decision method for online EEG processing in motor imagery BCI. *Intelligent Systems Design and Applications, 2010*.

Xu, W., Guan, C., Siong, C. E., Ranganatha, S., Thulasidas, M., & Wu, J. (2004). High accuracy classification of EEG signal. *Proceedings of the International Conference on Pattern Recognition, 2004*.

Yamagutchi, T., Nagata, K., Pham Quang, T., Pfurtscheller, G., & Inoue, K. (2007). Pattern recognition of EEG Signal during motor imagery by using SOM. *Proceedings of the Second International Conference on Innovative Computing, Information and Control, 2007*, 121-121.

Yin, K., Wu, J., & Zhang, J.-C. (2008). A framework of common spatial patterns based on support vector decomposition machine. *International Conference on Machine Learning and Cybernetics, 2008*.

Zenko, B., Todorovski, L., & Dzeroski, S. (2001). A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods. *Proceedings of the IEEE International Conference on Data Mining, 1*, 669-670.

Zhang, L., Liu, G., & Wu, Y. (2010). Wavelet and common spatial pattern for EEG signal feature extraction and classification. *Proceedings of the Conference on Computer, Mechatronics, Control and Electronic Engineering, 5*.

Zheng, X., Zhang, Q., Zhang, S., Yu, Y., Chen, W., Zhao, Y., & Lin, S. (2010). Real-time decoding algorithm in brain machine interfaces. *Proceedings of the IEEE/ICME International Conference on Complex Medical Engineering, 3*, 79-84.

Zocchi, C., Rovetta, A., & Fanfulla, F. (2007). Physiological parameters variation during driving simulations. *Proceedings of the IEEE/ASME Conference on Advanced Intelligent Mechatronics, 6*, 1-6.

APPENDIX A: COLLECTED RESULTS

Results for Cross Validation on Artificial Data

Results for Single Classifier Cross Validation on Artificial Data

SNR	16.00	3.00	1.00	0.30	0.03
Type					
Unbalanced					
LDA-CSP					
Acc	0.84	0.83	0.79	0.81	0.77
Sens	0.25	0.48	0.27	0.42	0.17
Spec	0.86	0.83	0.80	0.81	0.78
PPV	0.04	0.07	0.03	0.05	0.02
Phi	0.05	0.13	0.03	0.08	-0.02
LDA-PCA					
Acc	1.00	1.00	0.96	0.96	0.97
Sens	1.00	0.88	0.13	0.02	0.00
Spec	1.00	1.00	0.97	0.98	0.99
PPV	1.00	0.86	0.06	0.01	0.00
Phi	1.00	0.86	0.06	0.00	0.00
LDA-ADEN					
Acc	1.00	1.00	1.00	1.00	0.95
Sens	1.00	0.88	0.85	0.92	0.02
Spec	1.00	1.00	1.00	1.00	0.97
PPV	1.00	0.88	0.84	0.97	0.02
Phi	1.00	0.87	0.85	0.94	0.00
LDA-PCACSP					
Acc	1.00	1.00	0.96	0.92	0.97
Sens	1.00	0.88	0.13	0.04	0.00
Spec	1.00	1.00	0.97	0.94	0.99
PPV	1.00	0.86	0.06	0.00	0.00
Phi	1.00	0.86	0.06	-0.02	-0.01

RBF-CSP					
Acc	0.98	0.98	0.98	0.98	0.98
Sens	0.00	0.00	0.00	0.00	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	0.00	0.00	0.00	0.00	0.00
Phi	0.00	0.00	0.00	0.00	0.00
RBF-PCA					
Acc	1.00	1.00	0.89	0.78	0.98
Sens	1.00	0.88	0.15	0.15	0.00
Spec	1.00	1.00	0.90	0.79	1.00
PPV	1.00	0.86	0.05	0.01	0.00
Phi	1.00	0.86	0.05	-0.02	0.00
RBF-ADEN					
Acc	1.00	1.00	1.00	0.99	0.97
Sens	1.00	0.90	0.85	0.90	0.00
Spec	1.00	1.00	1.00	1.00	0.99
PPV	1.00	0.91	0.84	0.93	0.00
Phi	1.00	0.90	0.85	0.90	-0.01
RBF-PCACSP					
Acc	1.00	1.00	0.95	0.97	0.98
Sens	1.00	0.88	0.15	0.02	0.00
Spec	1.00	1.00	0.97	0.99	1.00
PPV	1.00	0.86	0.05	0.01	0.00
Phi	1.00	0.86	0.06	0.00	0.00
SVMG-CSP					
Acc	0.98	0.98	0.98	0.98	0.98
Sens	0.00	0.00	0.00	0.00	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	0.00	0.00	0.00	0.00	0.00
Phi	0.00	0.00	0.00	0.00	0.00
SVMG-					

PCA					
Acc	1.00	1.00	0.96	0.98	0.98
Sens	1.00	0.88	0.08	0.00	0.00
Spec	1.00	1.00	0.98	1.00	1.00
PPV	1.00	0.86	0.04	0.00	0.00
Phi	1.00	0.86	0.04	0.00	0.00
SVMG-ADEN					
Acc	1.00	1.00	1.00	1.00	0.98
Sens	1.00	0.88	0.88	0.94	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	1.00	0.88	0.84	0.97	0.00
Phi	1.00	0.88	0.86	0.95	0.00
SVMG-PCACSP					
Acc	1.00	1.00	0.96	0.98	0.98
Sens	1.00	0.88	0.08	0.00	0.00
Spec	1.00	1.00	0.98	1.00	1.00
PPV	1.00	0.86	0.04	0.00	0.00
Phi	1.00	0.86	0.04	0.00	0.00
SVMP-CSP					
Acc	0.98	0.98	0.98	0.98	0.98
Sens	0.00	0.00	0.00	0.00	0.02
Spec	1.00	1.00	1.00	1.00	1.00
PPV	0.00	0.00	0.00	0.00	0.03
Phi	0.00	0.00	0.00	0.00	0.02
SVMP-PCA					
Acc	1.00	1.00	0.96	0.98	0.98
Sens	1.00	0.88	0.08	0.00	0.00
Spec	1.00	1.00	0.98	1.00	1.00
PPV	1.00	0.86	0.04	0.00	0.00
Phi	1.00	0.86	0.04	0.00	0.00
SVMP-					

ADEN					
Acc	1.00	1.00	1.00	1.00	0.98
Sens	1.00	0.88	0.86	0.90	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	1.00	0.88	0.84	0.97	0.00
Phi	1.00	0.87	0.86	0.93	0.00
SVMP-PCACSP					
Acc	1.00	1.00	0.98	0.98	0.98
Sens	1.00	0.88	0.00	0.00	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	1.00	0.88	0.00	0.00	0.00
Phi	1.00	0.86	0.00	0.00	0.00
LDA-GADEN10					
Acc	1.00	1.00	0.99	1.00	0.96
Sens	1.00	1.00	0.85	0.98	0.00
Spec	1.00	1.00	1.00	1.00	0.98
PPV	1.00	0.97	0.86	0.94	0.00
Phi	1.00	0.98	0.85	0.96	-0.02

Mean Phis

SNR = 0.03 Artificial Events

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.00	-0.03	0.01	0.04	0.03	0.03	0.02	0.00	-0.02	0.02
ADENZ	-0.02	0.00	0.01	0.00	-0.01	-0.03	-0.01	-0.03	-0.02	-0.02
PCA	-0.01	-0.01	0.00	-0.01	-0.01	0.02	0.00	0.00	-0.01	-0.01
PLS	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Min

ADEN	-0.05	-0.09	-0.07	-0.04	0.00	-0.02	-0.07	-0.04	-0.07	-0.06
ADENZ	-0.04	-0.03	-0.04	-0.03	-0.04	-0.05	-0.07	-0.07	-0.06	-0.08
PCA	-0.05	-0.03	-0.03	-0.02	-0.01	-0.02	-0.01	-0.01	-0.01	-0.01
PLS	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02

Max

ADEN	0.06	0.04	0.08	0.14	0.10	0.09	0.10	0.05	0.03	0.09
ADENZ	-0.02	0.19	0.23	0.15	0.06	0.08	0.04	0.04	0.04	0.05
PCA	0.06	0.11	0.15	0.00	0.00	0.22	0.00	0.00	0.00	0.00
PLS	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06

SNR = 0.3 Artificial Events

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.94	0.95	0.93	0.92	0.93	0.93	0.91	0.89	0.89	0.72
ADENZ	0.88	0.76	0.78	0.74	0.67	0.75	0.76	0.74	0.73	0.66
PCA	0.03	0.07	0.28	0.28	0.28	0.25	0.25	0.25	0.25	0.25
PLS	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Min

ADEN	0.77	0.81	0.70	0.67	0.65	0.67	0.57	0.50	0.58	0.15
ADENZ	0.54	-0.01	-0.01	-0.01	-0.01	-0.01	0.22	-0.01	0.32	0.22
PCA	-0.06	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
PLS	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13

Max

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PCA	0.25	0.57	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PLS	0.05	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11

SNR = 1 Artificial Events

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.88	0.84	0.97	0.98	0.96	0.86	0.91	0.95	0.94	0.90

ADENZ	0.98	0.98	0.98	0.96	0.94	0.92	0.87	0.87	0.87	0.86
PCA	0.70	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
PLS	0.76	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78

Min

ADEN	0.70	0.40	0.77	0.81	0.77	0.32	0.45	0.61	0.66	0.50
ADENZ	0.86	0.86	0.86	0.66	0.54	0.35	-0.02	-0.01	-0.01	-0.01
PCA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PLS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Max

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PCA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PLS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

SNR = 3 Artificial Events

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.93	0.94	0.91
PCA	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
PLS	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Min

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.91
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	0.81	0.81	0.81	0.70
PCA	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
PLS	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91

Max

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PCA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PLS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

SNR = 16 Artificial Events

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.93	0.94	0.91
PCA	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
PLS	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Min

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.91

ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	0.81	0.81	0.81	0.70
PCA	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
PLS	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91

Max

ADEN	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ADENZ	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PCA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PLS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Results for Ensemble Classifiers on Artificial Data

SNR	16.00	3.00	1.00	0.30	0.03
Type					
Unbalanced					
Stacking					
LDA-ADEN					
Acc	1.00	1.00	0.99	0.99	0.27
Sens	1.00	1.00	1.00	0.98	0.73
Spec	1.00	1.00	0.99	0.99	0.26
PPV	1.00	1.00	0.99	0.99	0.27
Phi	0.95	0.96	0.86	0.84	0.00
Boosting					
LDA-ADEN					
Acc	0.98	0.98	0.98	0.98	0.98
Sens	0.04	0.00	0.00	0.00	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	0.13	0.00	0.00	0.00	0.00
Phi	0.07	0.00	0.00	0.00	0.00
Bagging					
LDA-ADEN					
Acc	1.00	1.00	1.00	1.00	0.98
Sens	1.00	1.00	0.75	0.50	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	1.00	1.00	0.75	0.50	0.00
Phi	1.00	1.00	0.75	0.50	0.00
Adaboost (3 weak learners)					
LDA-ADEN					
Acc	1.00	1.00	1.00	1.00	0.98
Sens	1.00	1.00	0.88	0.94	0.00
Spec	1.00	1.00	1.00	1.00	1.00
PPV	0.94	0.94	0.84	0.95	0.00
Phi	0.96	0.97	0.86	0.94	0.00

Results for Within Subject Phi and Study Informatics

Study A Subjects

Number	BMs	Duration	WS Phi	Literate
804	232	6.77	0.33	Literate
809	95	3.99	0.50	Literate
810	31	0.81	0.02	Threshold 1 (<.1)
811	68	1.12	0.14	Threshold 2 (<.15)
814	50	3.15	0.57	Literate
817	132	1.56	0.22	Literate
819	227	1.59	0.29	Literate
820	223	4.18	0.36	Literate
Mean	132.25	2.90	0.30	

Study C Subjects

Number	BMs	Duration	WS Phi	Literacy	Age	Sex
203	79	3.73	0.33	Literate	29	M
207	142	2.47	0.05	Threshold 1 (<.1)	27	M
208	75	5.05	0.22	Literate	30	F
210	36	6.30	0.22	Literate	23	M
211	105	5.45	0.53	Literate	24	M
213	44	5.02	0.50	Literate	33	M
214	73	1.91	-0.03	Threshold 1 (<.1)	41	M
216	80	2.37	-0.03	Threshold 1 (<.1)	45	F
217	68	1.34	0.12	Threshold 2 (<.15)	30	F
220	188	2.79	-0.25	Threshold 1 (<.1)	22	F
Mean	89.00	3.64	0.17		30.40	

Mean Phi Results on LOOCV with Single LDA Classifier

Mean Phis

SABIS LOOCV

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.23	0.19	0.21	0.22	0.24	0.23	0.21	0.19	0.21	0.19
ADENZ	0.27	0.23	0.18	0.22	0.13	0.17	0.10	0.12	0.12	0.15
PCA	0.27	0.26	0.26	0.24	0.25	0.24	0.25	0.24	0.23	0.23
PLS	0.18	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Min										
ADEN	0.06	0.03	-0.01	-0.01	0.03	0.01	0.00	-0.01	0.02	0.01
ADENZ	0.00	0.02	0.06	0.03	0.01	0.03	0.02	0.04	0.00	0.01
PCA	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.02	0.01	0.01
PLS	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Max										
ADEN	0.50	0.37	0.39	0.41	0.51	0.54	0.48	0.37	0.46	0.38
ADENZ	0.51	0.46	0.26	0.39	0.24	0.30	0.19	0.31	0.26	0.36
PCA	0.51	0.49	0.45	0.47	0.48	0.50	0.52	0.52	0.52	0.52
PLS	0.36	0.37	0.37	0.38	0.38	0.38	0.38	0.38	0.38	0.38

SCRIS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.04	0.10	0.00	0.01	0.00	0.01	-0.02	-0.01	0.03	0.01
ADENZ	-0.02	-0.03	0.00	-0.02	-0.02	-0.01	-0.03	-0.03	-0.03	-0.04
PCA	-0.04	-0.04	-0.04	-0.06	-0.08	-0.07	-0.06	-0.06	-0.04	-0.05
PLS	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Min										
ADEN	-0.04	0.00	-0.16	-0.08	-0.07	-0.08	-0.17	-0.12	-0.06	-0.09
ADENZ	-0.11	-0.16	-0.12	-0.11	-0.16	-0.11	-0.16	-0.21	-0.14	-0.15
PCA	-0.31	-0.35	-0.38	-0.40	-0.41	-0.39	-0.33	-0.31	-0.26	-0.26
PLS	-0.11	-0.09	-0.10	-0.09	-0.08	-0.09	-0.09	-0.09	-0.09	-0.08

Max										
ADEN	0.12	0.32	0.29	0.19	0.11	0.16	0.15	0.12	0.28	0.29
ADENZ	0.12	0.05	0.17	0.06	0.20	0.18	0.17	0.08	0.14	0.10
PCA	0.09	0.10	0.24	0.07	0.06	0.03	0.06	0.06	0.07	0.05
PLS	0.30	0.19	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.16	0.21	0.26	0.24	0.23	0.23	0.23	0.24	0.23	0.24
ADENZ	0.16	0.18	0.18	0.18	0.17	0.17	0.18	0.18	0.18	0.18
PCA	0.15	0.19	0.17	0.17	0.16	0.17	0.16	0.16	0.15	0.15
PLS	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13

Min										
ADEN	0.02	0.06	0.05	0.01	0.04	0.03	0.04	0.05	0.04	0.05
ADENZ	0.01	0.04	0.05	0.03	0.04	0.02	0.03	0.03	0.03	0.05
PCA	0.02	0.03	0.02	0.00	-0.01	-0.02	-0.02	-0.02	-0.03	-0.04
PLS	-0.09	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08

Max										
ADEN	0.33	0.34	0.43	0.42	0.36	0.37	0.38	0.38	0.35	0.39
ADENZ	0.38	0.35	0.40	0.44	0.33	0.35	0.33	0.33	0.34	0.35
PCA	0.31	0.41	0.40	0.41	0.42	0.44	0.44	0.43	0.40	0.41
PLS	0.27	0.28	0.28	0.27	0.28	0.28	0.28	0.28	0.28	0.28

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.21	0.23	0.23	0.21	0.23	0.24	0.26	0.27	0.27	0.27
ADENZ	0.16	0.18	0.16	0.18	0.18	0.18	0.19	0.20	0.19	0.19
PCA	0.18	0.15	0.16	0.17	0.16	0.17	0.16	0.15	0.14	0.14
PLS	0.11	0.12	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11

Min

ADEN	0.05	0.04	0.05	0.03	0.01	0.02	0.01	0.02	0.01	0.02
ADENZ	0.03	0.05	0.03	0.03	0.05	0.05	0.04	0.06	0.04	0.04
PCA	0.04	0.03	0.04	0.04	0.03	0.03	0.02	0.02	0.02	0.03
PLS	-0.15	-0.12	-0.12	-0.12	-0.12	-0.12	-0.13	-0.12	-0.13	-0.12
Max										
ADEN	0.36	0.49	0.45	0.43	0.47	0.48	0.57	0.56	0.56	0.57
ADENZ	0.29	0.33	0.30	0.33	0.31	0.30	0.31	0.35	0.36	0.36
PCA	0.34	0.32	0.42	0.45	0.45	0.42	0.45	0.39	0.33	0.33
PLS	0.32	0.32	0.33	0.33	0.33	0.32	0.33	0.33	0.32	0.32

SABIL										
Mean	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.26	0.26	0.19	0.16	0.13	0.12	0.14	0.15	0.15	0.16
ADENZ	0.33	0.27	0.18	0.14	0.04	0.02	0.03	0.02	0.02	0.01
PCA	0.23	0.27	0.26	0.28	0.27	0.26	0.26	0.27	0.28	0.29
PLS	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Min										
ADEN	0.06	0.07	0.05	0.02	0.00	-0.02	0.00	-0.01	0.03	-0.01
ADENZ	0.12	0.10	0.08	0.00	-0.03	-0.03	-0.06	-0.06	-0.06	-0.07
PCA	0.04	0.09	0.06	0.07	0.04	0.03	0.03	0.03	0.03	0.05
PLS	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Max										
ADEN	0.46	0.54	0.32	0.50	0.38	0.49	0.46	0.49	0.47	0.55
ADENZ	0.52	0.45	0.35	0.28	0.13	0.08	0.11	0.09	0.09	0.11
PCA	0.49	0.53	0.51	0.58	0.62	0.62	0.64	0.67	0.68	0.67
PLS	0.35	0.35	0.35	0.39	0.35	0.35	0.35	0.35	0.35	0.35

Mean Phi Results on LOOCV with Bagging

Phis

SABIS Bagging

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.09	0.12	0.14	0.15	0.17	0.19	0.18	0.18	0.19	0.18
ADENZ	0.19	0.16	0.14	0.09	0.08	0.07	0.07	0.10	0.09	0.07
PCA	0.21	0.23	0.23	0.23	0.22	0.23	0.24	0.23	0.25	0.23
PLS	0.17	0.18	0.17	0.17	0.17	0.17	0.17	0.17	0.18	0.17
Min										
ADEN	0.00	-0.01	-0.01	0.01	0.01	-0.02	-0.02	-0.02	-0.01	0.01
ADENZ	-0.02	0.02	0.04	0.00	-0.02	0.00	0.00	0.02	0.00	-0.07
PCA	0.03	0.03	0.04	0.04	0.04	0.02	0.02	0.04	0.04	0.04
PLS	0.03	0.02	0.02	0.02	0.02	0.03	0.02	0.03	0.03	0.02
Max										
ADEN	0.37	0.28	0.31	0.35	0.31	0.41	0.37	0.35	0.40	0.36
ADENZ	0.40	0.31	0.36	0.24	0.18	0.24	0.14	0.21	0.22	0.32
PCA	0.46	0.37	0.35	0.38	0.38	0.40	0.40	0.38	0.40	0.40
PLS	0.39	0.41	0.40	0.40	0.41	0.39	0.40	0.40	0.44	0.42

SCRIS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.02	0.00	0.03	0.07	0.05	0.09	0.04	0.10	0.06	0.04
ADENZ	-0.02	0.03	0.04	0.04	0.04	0.05	0.10	0.09	0.08	0.08
PCA	0.12	0.09	0.08	0.07	0.07	0.07	0.05	0.07	0.08	0.03
PLS	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03	0.04
Min										
ADEN	-0.07	-0.06	-0.02	-0.02	-0.05	0.01	-0.08	-0.04	-0.22	-0.11
ADENZ	-0.34	-0.15	-0.15	-0.23	-0.22	-0.12	-0.06	-0.08	-0.05	-0.06
PCA	-0.04	-0.06	-0.06	-0.07	-0.06	-0.07	-0.07	-0.06	-0.07	-0.06
PLS	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09

Max										
ADEN	0.22	0.03	0.21	0.33	0.33	0.36	0.34	0.35	0.35	0.33
ADENZ	0.10	0.11	0.20	0.20	0.13	0.22	0.32	0.34	0.30	0.29
PCA	0.38	0.39	0.37	0.31	0.30	0.31	0.31	0.32	0.32	0.31
PLS	0.17	0.25	0.21	0.25	0.24	0.24	0.24	0.25	0.21	0.26

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.15	0.18	0.18	0.20	0.21	0.21	0.21	0.22	0.21	0.25
ADENZ	0.15	0.17	0.17	0.17	0.19	0.20	0.20	0.20	0.21	0.21
PCA	0.17	0.16	0.20	0.19	0.20	0.19	0.19	0.19	0.19	0.19
PLS	0.13	0.12	0.12	0.13	0.13	0.12	0.13	0.12	0.13	0.12

Min										
ADEN	-0.01	0.00	0.02	0.01	0.00	0.01	0.01	0.03	0.03	0.04
ADENZ	0.02	0.05	0.04	0.05	0.06	0.06	0.06	0.07	0.08	0.08
PCA	0.06	0.01	0.05	0.05	0.06	0.04	0.04	0.05	0.06	0.06
PLS	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.00

Max										
ADEN	0.41	0.46	0.41	0.48	0.46	0.38	0.43	0.41	0.38	0.40
ADENZ	0.29	0.28	0.27	0.26	0.29	0.32	0.30	0.34	0.36	0.34
PCA	0.27	0.26	0.35	0.34	0.33	0.34	0.33	0.34	0.31	0.33
PLS	0.40	0.38	0.36	0.39	0.39	0.39	0.40	0.39	0.39	0.37

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.05	0.14	0.14	0.15	0.16	0.15	0.18	0.15	0.16	0.18
ADENZ	0.16	0.17	0.18	0.18	0.18	0.18	0.19	0.19	0.19	0.19
PCA	0.23	0.22	0.20	0.22	0.20	0.21	0.21	0.22	0.21	0.22
PLS	0.13	0.13	0.13	0.13	0.14	0.13	0.13	0.13	0.13	0.13

Min										
ADEN	-0.12	-0.09	-0.07	-0.07	0.01	0.00	0.04	-0.02	0.02	0.03
ADENZ	0.08	0.08	0.06	0.08	0.05	0.05	0.05	0.05	0.05	0.03

PCA	0.07	-0.02	0.04	0.04	0.03	0.05	0.05	0.05	0.06	0.07
PLS	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.04	0.03	0.04
Max										
ADEN	0.25	0.28	0.28	0.25	0.26	0.29	0.29	0.27	0.26	0.29
ADENZ	0.27	0.26	0.26	0.28	0.26	0.28	0.28	0.31	0.31	0.31
PCA	0.38	0.43	0.41	0.44	0.36	0.39	0.40	0.39	0.34	0.36
PLS	0.37	0.38	0.37	0.37	0.38	0.38	0.38	0.37	0.37	0.37
SABIL										
Mean	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.26	0.26	0.19	0.16	0.13	0.12	0.14	0.15	0.15	0.16
ADENZ	0.33	0.27	0.18	0.14	0.04	0.02	0.03	0.02	0.02	0.01
PCA	0.23	0.27	0.26	0.28	0.27	0.26	0.26	0.27	0.28	0.29
PLS	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Min										
ADEN	0.06	0.07	0.05	0.02	0.00	-0.02	0.00	-0.01	0.03	-0.01
ADENZ	0.12	0.10	0.08	0.00	-0.03	-0.03	-0.06	-0.06	-0.06	-0.07
PCA	0.04	0.09	0.06	0.07	0.04	0.03	0.03	0.03	0.03	0.05
PLS	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Max										
ADEN	0.46	0.54	0.32	0.50	0.38	0.49	0.46	0.49	0.47	0.55
ADENZ	0.52	0.45	0.35	0.28	0.13	0.08	0.11	0.09	0.09	0.11
PCA	0.49	0.53	0.51	0.58	0.62	0.62	0.64	0.67	0.68	0.67
PLS	0.35	0.35	0.35	0.39	0.35	0.35	0.35	0.35	0.35	0.35

Mean Phi Results on LOOCV with Mixed Data

Phis

SABIS Mixed										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.16	0.27	0.29	0.30	0.31	0.31	0.31	0.35	0.35	0.34
ADENZ	0.14	0.23	0.19	0.27	0.08	0.23	0.15	0.28	0.27	0.14
PCA	0.30	0.43	0.48	0.51	0.51	0.53	0.55	0.56	0.57	0.57
PLS	0.31	0.30	0.30	0.31	0.31	0.30	0.30	0.31	0.30	0.32
Min										
ADEN	0.15	0.25	0.26	0.29	0.29	0.29	0.28	0.33	0.34	0.32
ADENZ	0.01	0.12	0.09	0.15	-0.04	0.09	-0.04	0.22	0.20	0.00
PCA	0.29	0.41	0.45	0.50	0.50	0.52	0.54	0.53	0.55	0.56
PLS	0.30	0.29	0.29	0.30	0.29	0.27	0.30	0.31	0.29	0.31
Max										
ADEN	0.17	0.29	0.32	0.31	0.32	0.33	0.33	0.37	0.37	0.35
ADENZ	0.36	0.37	0.34	0.37	0.20	0.33	0.28	0.31	0.36	0.31
PCA	0.31	0.46	0.49	0.52	0.52	0.54	0.55	0.57	0.59	0.58
PLS	0.32	0.31	0.31	0.33	0.32	0.31	0.32	0.32	0.30	0.32
SCRIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.50	0.55	0.56	0.57	0.65	0.65	0.68	0.70	0.70	0.71
ADENZ	0.58	0.62	0.64	0.64	0.66	0.66	0.68	0.69	0.68	0.68
PCA	0.42	0.58	0.63	0.66	0.66	0.67	0.69	0.69	0.69	0.70
PLS	0.38	0.39	0.39	0.37	0.40	0.39	0.39	0.40	0.39	0.39
Min										
ADEN	0.49	0.55	0.56	0.56	0.64	0.64	0.67	0.70	0.70	0.70
ADENZ	0.58	0.61	0.63	0.63	0.65	0.65	0.67	0.68	0.67	0.67
PCA	0.40	0.57	0.62	0.64	0.64	0.66	0.68	0.67	0.68	0.68
PLS	0.37	0.38	0.38	0.37	0.38	0.39	0.38	0.39	0.37	0.38
Max										

ADEN	0.51	0.56	0.57	0.58	0.66	0.67	0.70	0.71	0.71	0.71
ADENZ	0.59	0.63	0.65	0.65	0.66	0.67	0.69	0.70	0.69	0.69
PCA	0.44	0.59	0.63	0.66	0.67	0.69	0.69	0.70	0.71	0.71
PLS	0.39	0.40	0.40	0.39	0.41	0.39	0.39	0.41	0.40	0.40

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.29	0.29	0.35	0.35	0.37	0.37	0.41	0.44	0.43	0.46
ADENZ	0.25	0.28	0.29	0.34	0.39	0.39	0.40	0.41	0.43	0.43
PCA	0.26	0.30	0.34	0.35	0.38	0.39	0.39	0.41	0.43	0.44
PLS	0.24	0.23	0.22	0.23	0.23	0.23	0.24	0.23	0.23	0.24

Min

ADEN	0.28	0.28	0.33	0.34	0.36	0.35	0.40	0.42	0.42	0.45
ADENZ	0.24	0.26	0.28	0.33	0.37	0.38	0.38	0.41	0.42	0.42
PCA	0.25	0.28	0.33	0.35	0.37	0.39	0.38	0.40	0.42	0.41
PLS	0.22	0.23	0.21	0.21	0.22	0.22	0.22	0.22	0.22	0.22

Max

ADEN	0.30	0.31	0.37	0.36	0.38	0.38	0.42	0.45	0.44	0.46
ADENZ	0.26	0.29	0.30	0.35	0.40	0.40	0.41	0.42	0.44	0.45
PCA	0.27	0.31	0.35	0.36	0.39	0.40	0.41	0.42	0.44	0.44
PLS	0.25	0.25	0.23	0.24	0.24	0.25	0.26	0.24	0.24	0.25

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.27	0.29	0.33	0.31	0.32	0.35	0.37	0.39	0.40	0.41
ADENZ	0.22	0.28	0.34	0.35	0.37	0.38	0.40	0.39	0.39	0.41
PCA	0.26	0.29	0.31	0.34	0.37	0.38	0.39	0.39	0.41	0.42
PLS	0.25	0.25	0.24	0.25	0.24	0.24	0.24	0.25	0.24	0.25

Min

ADEN	0.26	0.29	0.31	0.30	0.30	0.32	0.36	0.38	0.39	0.39
ADENZ	0.21	0.28	0.33	0.34	0.36	0.37	0.39	0.39	0.38	0.40
PCA	0.24	0.27	0.30	0.34	0.37	0.37	0.38	0.38	0.40	0.41

PLS	0.24	0.24	0.24	0.24	0.23	0.23	0.24	0.24	0.24	0.24
Max										
ADEN	0.28	0.30	0.35	0.31	0.33	0.37	0.39	0.41	0.40	0.42
ADENZ	0.24	0.29	0.35	0.36	0.39	0.38	0.41	0.40	0.40	0.43
PCA	0.28	0.31	0.32	0.36	0.38	0.39	0.39	0.40	0.42	0.44
PLS	0.25	0.26	0.25	0.27	0.25	0.25	0.26	0.26	0.25	0.26
SABIL	Features									
Mean	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.29	0.37	0.42	0.47	0.51	0.47	0.48	0.51	0.52	0.52
ADENZ	0.36	0.17	0.11	0.11	0.11	0.12	0.12	0.12	0.13	0.13
PCA	0.26	0.40	0.47	0.49	0.51	0.52	0.53	0.55	0.56	0.55
PLS	0.21	0.23	0.29	0.32	0.27	0.27	0.31	0.28	0.28	0.27
Min										
ADEN	0.28	0.35	0.41	0.45	0.49	0.46	0.46	0.48	0.50	0.51
ADENZ	0.35	0.04	0.01	0.01	-0.01	-0.01	0.01	0.01	0.00	0.01
PCA	0.25	0.39	0.45	0.48	0.49	0.50	0.52	0.53	0.55	0.53
PLS	0.20	0.22	0.28	0.29	0.26	0.26	0.29	0.27	0.27	0.25
Max										
ADEN	0.29	0.38	0.44	0.49	0.52	0.48	0.49	0.54	0.54	0.54
ADENZ	0.37	0.26	0.15	0.15	0.14	0.16	0.16	0.16	0.17	0.16
PCA	0.27	0.41	0.48	0.51	0.52	0.53	0.53	0.56	0.57	0.57
PLS	0.21	0.24	0.30	0.34	0.28	0.27	0.32	0.29	0.29	0.27

SABUL	Features									
Mean	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.27	0.28	0.26	0.18	0.16	0.15	0.13	0.13	0.14	0.13
ADENZ	0.27	0.20	0.15	0.07	-0.01	-0.01	0.00	-0.01	0.00	0.00
PCA	0.20	0.22	0.22	0.21	0.23	0.22	0.21	0.22	0.23	0.23
PLS	0.16	0.15	0.14	0.15	0.15	0.15	0.15	0.15	0.15	0.15
Min										
ADEN	0.05	0.05	0.06	0.03	0.03	-0.02	-0.04	-0.04	-0.03	-0.03
ADENZ	0.10	0.05	0.06	-0.08	-0.11	-0.08	-0.04	-0.04	-0.02	-0.01
PCA	0.04	0.06	0.04	0.02	0.02	0.04	0.01	0.02	0.02	0.02
PLS	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Max										
ADEN	0.57	0.56	0.46	0.51	0.44	0.47	0.42	0.57	0.63	0.63
ADENZ	0.43	0.34	0.26	0.16	0.07	0.04	0.05	0.03	0.04	0.02
PCA	0.53	0.54	0.63	0.62	0.66	0.68	0.68	0.69	0.68	0.68
PLS	0.29	0.29	0.28	0.28	0.29	0.29	0.29	0.29	0.29	0.29

Mean Phi Results on LOOCV with Pre-Onset Periods

Mean Phi Results on LOOCV with 1-s Pre-Onset Periods with Events

Phis

SABIS	1 sec onset									
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.26	0.21	0.21	0.26	0.25	0.21	0.19	0.21	0.20	0.22
ADENZ	0.23	0.22	0.15	0.11	0.17	0.15	0.13	0.11	0.15	0.10
PCA	0.27	0.27	0.27	0.26	0.26	0.26	0.25	0.25	0.24	0.24
PLS	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
Min										
ADEN	-0.01	0.05	0.02	0.04	0.03	0.00	0.01	0.05	0.04	0.01
ADENZ	0.07	0.06	0.04	0.00	0.04	0.04	0.05	0.02	0.03	0.02
PCA	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.01	0.03	0.03
PLS	0.02	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Max										
ADEN	0.54	0.41	0.33	0.52	0.52	0.49	0.40	0.35	0.41	0.45
ADENZ	0.38	0.45	0.27	0.17	0.29	0.34	0.24	0.24	0.35	0.26
PCA	0.49	0.49	0.47	0.46	0.47	0.47	0.47	0.48	0.48	0.49
PLS	0.37	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37

SCRIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.06	0.09	0.03	0.01	0.01	0.00	-0.01	-0.03	0.01	0.02
ADENZ	0.03	0.02	-0.01	-0.02	-0.04	-0.02	-0.03	0.00	-0.02	-0.01
PCA	-0.03	-0.04	-0.04	-0.06	-0.07	-0.06	-0.06	-0.07	-0.04	-0.06
PLS	0.06	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Min										
ADEN	-0.05	-0.02	-0.16	-0.13	-0.11	-0.13	-0.24	-0.22	-0.07	-0.09
ADENZ	-0.07	-0.12	-0.11	-0.17	-0.18	-0.11	-0.11	-0.09	-0.16	-0.09
PCA	-0.27	-0.33	-0.35	-0.38	-0.38	-0.36	-0.31	-0.31	-0.26	-0.25
PLS	-0.07	-0.11	-0.08	-0.09	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09

Max										
ADEN	0.22	0.23	0.24	0.12	0.13	0.16	0.22	0.19	0.26	0.26
ADENZ	0.12	0.13	0.14	0.09	0.04	0.18	0.08	0.10	0.15	0.16
PCA	0.07	0.10	0.18	0.07	0.06	0.02	0.03	0.05	0.04	0.03
PLS	0.21	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.17	0.25	0.27	0.24	0.25	0.23	0.24	0.24	0.23	0.24
ADENZ	0.20	0.20	0.18	0.19	0.18	0.19	0.19	0.19	0.18	0.18
PCA	0.16	0.19	0.18	0.18	0.17	0.17	0.16	0.16	0.16	0.15
PLS	0.13	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
Min										
ADEN	0.04	0.07	0.06	0.02	0.05	0.06	0.07	0.05	0.04	0.03
ADENZ	0.06	0.07	0.07	0.05	0.04	0.03	0.04	0.05	0.03	0.03
PCA	0.03	0.04	0.03	0.02	0.01	0.00	-0.01	-0.02	-0.02	-0.03
PLS	-0.06	-0.05	-0.04	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
Max										
ADEN	0.33	0.39	0.41	0.39	0.37	0.34	0.37	0.38	0.36	0.35
ADENZ	0.33	0.34	0.33	0.34	0.38	0.36	0.35	0.34	0.31	0.31
PCA	0.34	0.37	0.35	0.37	0.39	0.40	0.39	0.37	0.36	0.37
PLS	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.22	0.27	0.24	0.23	0.24	0.26	0.29	0.29	0.30	0.30
ADENZ	0.18	0.20	0.18	0.18	0.19	0.18	0.19	0.19	0.20	0.21
PCA	0.19	0.17	0.18	0.18	0.18	0.18	0.17	0.17	0.15	0.15
PLS	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Min										
ADEN	0.04	0.07	0.06	0.03	0.03	0.06	0.06	0.03	0.03	0.02

ADENZ	0.05	0.07	0.05	0.02	0.07	0.05	0.03	0.01	0.02	0.03
PCA	0.05	0.06	0.07	0.06	0.04	0.04	0.03	0.05	0.03	0.03
PLS	-0.08	-0.03	-0.02	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02
Max										
ADEN	0.36	0.50	0.46	0.43	0.45	0.49	0.55	0.53	0.53	0.55
ADENZ	0.29	0.32	0.29	0.29	0.31	0.30	0.32	0.32	0.32	0.35
PCA	0.34	0.33	0.36	0.40	0.40	0.34	0.38	0.36	0.28	0.28
PLS	0.33	0.33	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34

SABIL										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.21	0.24	0.23	0.16	0.18	0.15	0.15	0.15	0.17	0.14
ADENZ	0.32	0.25	0.20	0.05	0.05	-0.02	-0.01	0.00	-0.01	0.00
PCA	0.22	0.26	0.25	0.27	0.27	0.26	0.26	0.26	0.28	0.28
PLS	0.23	0.22	0.22	0.23	0.22	0.22	0.22	0.22	0.22	0.22
Min										
ADEN	0.05	0.10	0.10	0.06	0.03	0.01	0.00	0.02	0.01	0.01
ADENZ	0.13	0.07	0.09	-0.02	0.01	-0.11	-0.04	-0.07	-0.07	-0.02
PCA	0.07	0.09	0.06	0.09	0.09	0.08	0.04	0.04	0.05	0.03
PLS	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Max										
ADEN	0.38	0.39	0.42	0.45	0.48	0.54	0.56	0.56	0.58	0.50
ADENZ	0.43	0.38	0.34	0.10	0.11	0.06	0.02	0.06	0.04	0.03
PCA	0.43	0.46	0.48	0.53	0.57	0.58	0.62	0.63	0.65	0.65
PLS	0.35	0.33	0.37	0.38	0.36	0.36	0.36	0.36	0.36	0.36

Mean Phi Results on LOOCV with 1-s Predictive Case

Phis

SABIS 1 s onset prediction

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.05	0.03	0.03	0.02	0.04	0.02	0.02	0.03	0.03	0.02
ADENZ	0.02	0.05	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01
PCA	0.05	0.06	0.06	0.05	0.03	0.02	0.02	0.02	0.02	0.02
PLS	0.03	0.03	0.02	0.03	0.03	0.03	0.02	0.03	0.03	0.03
Min										
ADEN	-0.01	-0.01	-0.01	0.00	0.00	-0.01	-0.01	0.00	-0.01	-0.01
ADENZ	-0.01	0.01	-0.02	0.00	-0.02	-0.02	-0.03	-0.03	-0.01	0.00
PCA	0.01	0.02	0.02	0.03	0.01	0.01	0.00	0.01	0.00	0.00
PLS	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Max										
ADEN	0.14	0.07	0.07	0.06	0.08	0.07	0.05	0.09	0.11	0.07
ADENZ	0.05	0.12	0.04	0.05	0.04	0.04	0.04	0.03	0.05	0.03
PCA	0.08	0.09	0.09	0.08	0.07	0.05	0.04	0.04	0.04	0.04
PLS	0.09	0.10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09

SCRIS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.02	0.00	-0.01	-0.01	0.01	-0.01	0.00	0.01	0.01	0.00
ADENZ	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.01
PCA	0.00	0.00	0.00	-0.01	-0.01	0.00	-0.01	0.00	0.00	-0.01
PLS	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Min										
ADEN	-0.02	-0.07	-0.06	-0.04	-0.02	-0.12	-0.06	-0.05	-0.06	-0.05
ADENZ	-0.04	-0.03	-0.04	-0.03	-0.03	-0.03	-0.04	-0.04	-0.03	-0.03
PCA	-0.05	-0.03	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03	-0.02
PLS	-0.06	-0.05	-0.06	-0.07	-0.06	-0.07	-0.07	-0.07	-0.07	-0.07
Max										
ADEN	0.05	0.06	0.06	0.04	0.06	0.03	0.07	0.06	0.05	0.04

ADENZ	0.08	0.05	0.05	0.05	0.05	0.06	0.05	0.03	0.05	0.04
PCA	0.05	0.06	0.02	0.03	0.00	0.09	0.01	0.08	0.04	0.01
PLS	0.02	0.02	0.02	0.04	0.03	0.03	0.03	0.03	0.03	0.03

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.01	0.04	0.01	0.01	0.02	0.01	0.02	0.01	0.01	0.02
ADENZ	0.00	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.01	0.02
PCA	0.05	0.05	0.05	0.03	0.03	0.03	0.03	0.03	0.02	0.02
PLS	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Min										
ADEN	-0.01	0.00	-0.01	-0.01	-0.02	-0.01	-0.01	-0.01	-0.02	0.00
ADENZ	-0.02	-0.02	-0.01	-0.02	-0.01	-0.02	-0.01	-0.02	-0.01	-0.02
PCA	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00	0.00	0.00	-0.01
PLS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Max										
ADEN	0.04	0.11	0.05	0.06	0.08	0.05	0.05	0.04	0.04	0.05
ADENZ	0.03	0.04	0.06	0.07	0.06	0.06	0.07	0.08	0.06	0.07
PCA	0.14	0.15	0.16	0.13	0.10	0.08	0.07	0.08	0.08	0.07
PLS	0.09	0.11	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.00	0.02	0.03	0.03	0.03	0.03	0.02	0.02	0.03	0.02
ADENZ	0.02	0.01	0.01	0.02	0.01	0.01	0.02	0.02	0.02	0.02
PCA	0.05	0.05	0.05	0.05	0.03	0.04	0.04	0.03	0.03	0.03
PLS	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Min										
ADEN	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
ADENZ	-0.02	0.00	-0.02	-0.01	-0.01	-0.01	-0.01	-0.02	-0.01	-0.02
PCA	0.00	-0.01	-0.03	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	-0.01
PLS	-0.02	0.00	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Max										
ADEN	0.02	0.07	0.08	0.08	0.08	0.06	0.07	0.05	0.06	0.05

ADENZ	0.06	0.06	0.04	0.06	0.05	0.04	0.05	0.07	0.08	0.09
PCA	0.13	0.13	0.12	0.13	0.11	0.10	0.10	0.09	0.08	0.06
PLS	0.08	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09

SABIL										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.02	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00
ADENZ	0.05	0.05	0.00	0.00	-0.01	-0.01	0.01	0.00	0.00	0.01
PCA	0.03	0.04	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.03
PLS	0.05	0.05	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.04
Min										
ADEN	-0.03	-0.04	-0.01	-0.02	0.00	-0.01	0.00	0.00	0.00	0.00
ADENZ	0.01	0.00	-0.04	-0.03	-0.02	-0.02	-0.01	-0.02	-0.01	0.00
PCA	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.01	-0.01
PLS	-0.04	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
Max										
ADEN	0.06	0.05	0.02	0.03	0.09	0.04	0.01	0.06	0.00	0.00
ADENZ	0.09	0.09	0.04	0.02	0.02	0.01	0.03	0.03	0.01	0.04
PCA	0.05	0.10	0.09	0.10	0.09	0.09	0.09	0.09	0.10	0.10
PLS	0.15	0.15	0.12	0.13	0.15	0.15	0.15	0.15	0.15	0.15

Mean Phi Results on LOOCV with MISFETS

Initial MISFETS LOOCV Results

Phis	MISFETS									
SABIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.01	0.20	0.19	0.19	0.20	0.20	0.20	0.21	0.20	0.19
ADENZ	0.06	0.14	0.12	0.12	0.11	0.12	0.12	0.12	0.11	0.12
RADEN	0.22	0.26	0.25	0.23	0.20	0.20	0.19	0.21	0.19	0.20
RADENZ	0.28	0.19	0.21	0.13	0.12	0.12	0.08	0.10	0.07	0.06
Min										
ADEN	-0.11	0.00	0.01	0.01	0.01	0.02	0.03	0.02	0.02	0.02
ADENZ	-0.01	-0.01	0.02	0.04	0.03	0.01	0.02	0.00	0.02	0.03
RADEN	0.01	0.04	0.04	0.00	-0.01	0.00	0.01	0.00	0.02	0.02
RADENZ	0.01	0.04	0.04	0.00	-0.01	0.00	0.01	0.00	0.02	0.02
Max										
ADEN	0.26	0.45	0.51	0.52	0.53	0.44	0.43	0.40	0.40	0.36
ADENZ	0.29	0.29	0.26	0.26	0.20	0.21	0.22	0.21	0.23	0.25
RADEN	0.39	0.58	0.40	0.39	0.34	0.33	0.43	0.37	0.44	0.30
RADENZ	0.52	0.42	0.39	0.34	0.25	0.31	0.18	0.19	0.19	0.16

SCRIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Min										
ADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Max										
ADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADEN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RADENZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.18	0.17	0.17	0.17	0.21	0.21	0.21	0.22	0.22	0.23
ADENZ	0.18	0.15	0.18	0.16	0.18	0.13	0.14	0.14	0.15	0.13
RADEN	0.17	0.26	0.25	0.23	0.23	0.24	0.24	0.21	0.23	0.22
RADENZ	0.19	0.19	0.22	0.23	0.20	0.21	0.25	0.21	0.22	0.21

Min										
ADEN	0.06	0.07	0.03	-0.03	-0.01	-0.01	0.01	0.04	0.05	0.06
ADENZ	0.05	0.02	-0.01	0.00	0.01	0.01	0.00	0.02	0.03	-0.02
RADEN	-0.02	0.03	0.03	0.01	0.05	0.01	0.03	0.02	0.04	0.04
RADENZ	-0.02	0.03	0.03	0.01	0.05	0.01	0.03	0.02	0.04	0.04

Max										
ADEN	0.31	0.29	0.32	0.33	0.37	0.37	0.36	0.38	0.37	0.36
ADENZ	0.32	0.30	0.28	0.30	0.31	0.24	0.27	0.31	0.34	0.32
RADEN	0.37	0.41	0.46	0.43	0.43	0.44	0.44	0.31	0.48	0.34
RADENZ	0.44	0.41	0.42	0.44	0.38	0.43	0.47	0.37	0.40	0.39

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.16	0.17	0.18	0.21	0.22	0.23	0.22	0.23	0.24	0.23
ADENZ	0.16	0.22	0.08	0.13	0.02	0.01	0.03	0.02	0.04	0.06
RADEN	0.17	0.19	0.21	0.13	0.19	0.16	0.15	0.21	0.22	0.20
RADENZ	0.13	0.15	0.17	0.17	0.16	0.16	0.12	0.12	0.17	0.09

Min										
ADEN	-0.04	0.01	0.04	0.06	0.05	0.07	0.06	0.08	0.10	0.06

ADENZ	0.06	0.01	-0.14	-0.01	-0.06	-0.07	-0.04	-0.08	-0.06	-0.06
RADEN	0.00	0.03	0.10	-0.03	0.01	-0.05	0.02	0.05	0.09	0.01
RADENZ	0.00	0.03	0.10	-0.03	0.01	-0.05	0.02	0.05	0.09	0.01
Max										
ADEN	0.30	0.33	0.28	0.30	0.31	0.32	0.32	0.36	0.38	0.37
ADENZ	0.25	0.31	0.22	0.27	0.10	0.10	0.13	0.15	0.19	0.20
RADEN	0.28	0.34	0.33	0.29	0.35	0.34	0.27	0.35	0.41	0.36
RADENZ	0.24	0.28	0.32	0.30	0.40	0.40	0.26	0.25	0.34	0.27

Abridged Dataset LOOCV Results

Phis	Abridged									
SABIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.25	0.28	0.25	0.26	0.24	0.23	0.24	0.24	0.23	0.20
ADENZ	0.22	0.17	0.27	0.20	0.10	0.15	0.11	0.13	0.08	0.11
PCA	0.27	0.25	0.23	0.23	0.23	0.23	0.23	0.22	0.22	0.22
PLS	0.18	0.19	0.19	0.18	0.19	0.19	0.18	0.18	0.18	0.18
Min										
ADEN	0.01	0.04	0.02	0.02	0.01	-0.01	0.01	0.02	0.02	0.01
ADENZ	0.00	0.01	0.04	0.04	0.01	0.03	0.01	0.03	0.00	0.03
PCA	0.01	0.00	0.00	0.00	0.01	0.03	0.04	0.01	0.01	0.01
PLS	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Max										
ADEN	0.53	0.49	0.48	0.60	0.52	0.44	0.52	0.51	0.45	0.47
ADENZ	0.55	0.30	0.49	0.37	0.25	0.32	0.24	0.27	0.22	0.31
PCA	0.46	0.42	0.42	0.44	0.46	0.49	0.49	0.48	0.49	0.49
PLS	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
SCRIS										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.04	0.01	-0.02	0.01	0.02	-0.01	-0.06	-0.05	-0.03	-0.03
ADENZ	0.05	0.03	0.04	0.03	0.00	0.00	-0.02	-0.04	-0.01	-0.03
PCA	0.04	0.03	0.05	0.04	0.03	0.02	-0.01	0.01	-0.02	-0.04
PLS	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
Min										
ADEN	-0.06	-0.09	-0.35	-0.24	-0.04	-0.25	-0.25	-0.23	-0.18	-0.22
ADENZ	-0.02	-0.07	-0.06	-0.05	-0.15	-0.14	-0.16	-0.24	-0.18	-0.22
PCA	-0.09	-0.09	-0.09	-0.07	-0.09	-0.14	-0.15	-0.10	-0.12	-0.17
PLS	-0.16	-0.29	-0.30	-0.30	-0.29	-0.29	-0.29	-0.29	-0.29	-0.29
Max										
ADEN	0.23	0.20	0.25	0.14	0.14	0.07	0.05	0.08	0.08	0.12
ADENZ	0.21	0.11	0.24	0.10	0.11	0.12	0.08	0.08	0.08	0.06
PCA	0.34	0.18	0.34	0.29	0.26	0.29	0.12	0.19	0.07	0.04
PLS	0.05	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06

SARUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.20	0.25	0.26	0.25	0.27	0.26	0.26	0.25	0.24	0.23
ADENZ	0.21	0.21	0.20	0.21	0.21	0.23	0.23	0.22	0.22	0.22
PCA	0.20	0.21	0.20	0.21	0.20	0.20	0.20	0.21	0.21	0.21
PLS	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
Min										
ADEN	0.02	0.04	0.05	0.04	0.03	0.04	0.03	0.01	0.00	0.01
ADENZ	0.05	0.05	0.04	0.05	0.04	0.06	0.04	0.05	0.06	0.05
PCA	0.06	0.05	0.05	0.05	0.04	0.03	0.04	0.03	0.03	0.03
PLS	0.00	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Max										
ADEN	0.37	0.42	0.45	0.45	0.44	0.45	0.39	0.39	0.38	0.36
ADENZ	0.36	0.36	0.37	0.36	0.42	0.43	0.43	0.40	0.41	0.42
PCA	0.30	0.38	0.34	0.37	0.39	0.41	0.38	0.41	0.42	0.40
PLS	0.26	0.25	0.25	0.26	0.26	0.25	0.25	0.25	0.25	0.25

SABUS

Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.22	0.22	0.25	0.23	0.24	0.25	0.26	0.24	0.24	0.26
ADENZ	0.21	0.20	0.20	0.20	0.19	0.19	0.18	0.17	0.17	0.17
PCA	0.18	0.20	0.21	0.20	0.20	0.19	0.17	0.17	0.17	0.17
PLS	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Min										
ADEN	0.10	0.03	0.12	0.04	0.02	0.01	0.00	0.02	0.02	0.03
ADENZ	0.06	0.07	0.06	0.04	0.05	0.06	0.05	0.05	0.05	0.05
PCA	0.04	0.05	0.04	0.04	0.03	0.04	0.04	0.05	0.05	0.05
PLS	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Max										
ADEN	0.33	0.33	0.40	0.42	0.40	0.43	0.47	0.42	0.42	0.45
ADENZ	0.29	0.35	0.32	0.32	0.34	0.33	0.35	0.33	0.34	0.34
PCA	0.30	0.35	0.33	0.32	0.32	0.30	0.29	0.30	0.31	0.32
PLS	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

SABIL										
Features	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00	90.00	100.00
ADEN	0.28	0.31	0.31	0.29	0.13	0.12	0.13	0.14	0.12	0.13
ADENZ	0.33	0.20	0.14	0.13	0.12	0.07	0.00	0.00	-0.01	0.00
PCA	0.29	0.29	0.33	0.34	0.32	0.32	0.32	0.31	0.30	0.31
PLS	0.24	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
Min										
ADEN	0.10	0.11	0.08	0.08	0.01	0.01	0.00	0.02	0.04	0.04
ADENZ	0.12	0.06	0.07	0.08	0.02	-0.06	-0.02	-0.01	-0.03	-0.03
PCA	0.10	0.10	0.15	0.14	0.15	0.15	0.14	0.13	0.12	0.10
PLS	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Max										
ADEN	0.46	0.51	0.62	0.61	0.39	0.42	0.38	0.38	0.38	0.34
ADENZ	0.53	0.33	0.17	0.24	0.27	0.24	0.02	0.02	0.02	0.03
PCA	0.57	0.53	0.53	0.56	0.61	0.63	0.65	0.67	0.68	0.69
PLS	0.38	0.48	0.47	0.47	0.46	0.46	0.46	0.46	0.46	0.46

Results for WEKA Validation

Test		1	2	3	4	5	6	7	8			
PCA10	Stack									Mean	Min	Max
Acc		0.72	0.94	0.99	0.95	0.96	0.91	0.93	0.82	0.90	0.72	0.99
Sens		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Spec		0.72	0.94	0.99	0.95	0.96	0.91	0.93	0.82	0.90	0.72	0.99
PPV		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Phi		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PCA10	Adaboost											
Acc		0.92	0.37	0.00	0.10	0.44	0.31	0.42	0.81	0.42	0.00	0.92
Sens		0.92	0.37	0.00	0.10	0.44	0.31	0.42	0.17	0.34	0.00	0.92
Spec		0.75	0.97	0.99	0.95	0.99	0.92	0.96	0.82	0.92	0.75	0.99
PPV		0.11	0.58	0.00	0.08	0.81	0.20	0.50	0.00	0.29	0.00	0.81
Phi		0.27	0.42	0.00	0.05	0.58	0.19	0.41	0.00	0.24	0.00	0.58
PCA10	Bagging											
Acc		0.76	0.42	0.00	0.03	0.17	0.34	0.33	0.81	0.36	0.00	0.81
Sens		0.76	0.42	0.00	0.03	0.17	0.34	0.33	0.22	0.28	0.00	0.76
Spec		0.75	0.96	0.99	0.95	0.97	0.94	0.96	0.82	0.92	0.75	0.99
PPV		0.14	0.37	0.00	0.04	0.37	0.45	0.46	0.01	0.23	0.00	0.46
Phi		0.25	0.35	-0.01	-0.03	0.21	0.32	0.33	0.01	0.18	-0.03	0.35

APPENDIX B: ICTOMI DOCUMENTATION

Integrated Canterbury Open Modular Inventory (ICTOMI) Toolset

v.3.1

By John LaRocco

The following toolset was written for the fulfilment of a PhD thesis at the University of Canterbury. The original intention was EEG signal processing, but it can be applied to other fields. The main files are included here. It was written on MATLAB R2010a. It has the signal processing, statistics, and machine learning toolboxes. There's also the SVM-KM toolbox, EEGLAB, and the Speech Analysis toolbox (http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/doc/index.html) required to run some programs.

1. Updates

V.3.1 Updates:

-Added mixing and bagging cross-validation.

V.3.0 Updates:

-Malik Style Single Classification Modules (single classifier and stacking ensembles added).

-ADEN, PLS, GA, and GADEN feature reduction/selection modules added.

-CSP modules dropped.

-AdaBoost added.

-Automated preprocessing functions for Study 2 data added.

-Within-subject nescience data for Peiris Data and Poudel Study II included.

V.2.0 Updates:

-SOM Pattern Recognition modules dropped.

-CSPPCA Feature Selection modules dropped.

-PPV and NPV added to standard performance outputs, as well as a "failsafe" method of calculating accuracy.

2. Dummy Data Modules

Dummy Data: Utility modules that generate fake data for testing and validation.

-Random Noise Dummy Data: Datasets consist of junk data and labels. Complete and utter random noise. There is nothing to learn here. The best theoretical performance you can get is 50% accuracy. Inputs are the total time you desire in seconds (*finaltime*), number of subjects (*subs*), and number of channels (*chans*). The outputs are cells of dummy data (*total_segments*) and target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into channel by time domain data by instances. For target labels, the number of cells are equal to the number of subjects, and the dimensions of each cell are 1 by instances.

[total_segments,total_labels]=prototype_dummydata_gen_mark2(finaltime,subs,chans);

-Gaussian Dummy Data: Gaussian Dummy data is two separate Gaussian distributions with proper group labels. Inputs are the number of features (features), number of instances (instances), and number of subjects (subs). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells are equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are balanced, with an equal number of each class present.

[total_segments,total_labels]=toydata_gen(features,instances,subs);

-Spectral Dummy Data: Gaussian Dummy data is two separate sinusoidal signals with Gaussian noise with proper group labels. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are balanced, with an equal number of each class present.

[total_segments,total_labels]=toydata_gen_waves(features,instances,subs);

-Hard Dummy Data: The dummy data is separated into two equally balanced classes. Each class is a sum of different periodic sinusoidal signals. The noise is randomly generated Gaussian noise. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are balanced, with an equal number of each class present.

[total_segments,total_labels]=dummydata_hard(features,instances,subs);

-Easy Dummy Data: The dummy data is separated into two equally balanced classes. Each class is a sum of different periodic sinusoidal signals. The noise is randomly generated Gaussian noise, but less noise than in the "hard" data. There is also an offset value between both classes. Inputs are the number of features (*features*), number of instances (*instances*),

and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are balanced, with an equal number of each class present.

[total_segments,total_labels]=dummydata_easy(features,instances,subs);

-Noise Dummy Data: The noise dummy data is separated into two equally balanced classes. The noise in each class is no different than the other class. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are balanced, with an equal number of each class present.

[total_segments,total_labels]=dummydata_noise(features,instances,subs);

-Unbalanced Hard Dummy Data: The dummy data is separated into two unbalanced classes. Each class is a sum of different periodic sinusoidal signals. The noise is randomly generated Gaussian noise. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are unbalanced, with 25% of the data belonging to one class and 75% belonging to the other. The division of data can be modified by manually changing the variable (*midpoint*).

[total_segments,total_labels]=dummydata_hard_unbalanced (features,instances,subs);

-Unbalanced Easy Dummy Data: The dummy data is separated into two unbalanced classes. Each class is a sum of different periodic sinusoidal signals. The noise is randomly generated Gaussian noise, but less noise than in the "hard" data. There is also an offset value between both classes. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The datasets are unbalanced, with 25% of the data belonging to one class and 75% belonging to the other. The division of data can be modified by manually changing the variable (*midpoint*).

[total_segments,total_labels]=dummydata_easy_unbalanced (features,instances,subs);

-Unbalanced Noise Dummy Data: The noise dummy data is separated into two unbalanced classes. The noise in each class is no different than the other class. Inputs are the number of features (*features*), number of instances (*instances*), and number of subjects (*subs*). Outputs include are a struct of data (*total_segments*) and struct full of target labels (*total_labels*). The dimensions of the dummy data are one cell per simulated subject. The contents of each cell are arranged into time domain features by instances. For target labels, the number of cells is equal to the number of subjects, and the dimensions of each cell are 1 by instances. The

datasets are unbalanced, with 25% of the data belonging to one class and 75% belonging to the other. The division of data can be modified by manually changing the variable (*midpoint*).

[total_segments,total_labels]=dummydata_noise_unbalanced(features,instances,subs);

3. Utility Modules

Utility Modules: Utility modules and scripts handle preprocessing, support, validation, loading data, artifact removal, and other essential functions.

-Correction: Done: Evaluates performance and gives accuracy, phi, area under Receiver Operating Characteristic (ROC) curve, and other metrics. The input is the matrix of guesses from a classifier function (*ypred*) and the "answer key" to the testing data (*testing_label*). The required dimensions of the input are the matrix of classifier output and the "answer key" must be instances by one.

[phi,roc,auc_roc,accuracy,sensitivity,specificity,
acc2,ppv,npv]=prototype_correction(ypred,testing_label);

-Cleanup: Done: Removes NaNs and Infs from a matrix (*X*).

[X]=prototype_cleanup(X);

-Unify Channel: Done: Combines multiple channels worth of features into one. It is intended for data organized in the format channel by features by instances. The resulting matrix is organized into features by instances.

y=prototype_unify_channel(y);

-Stacking Cleanup: Done: Removes NaNs and Infs from a matrix (*X*). Also, the raw output of a stacking classifier ensemble (*outprobs*) is retained, before values are normalized. It should be used after a classification step.

[X, outprobs]=stacking_cleaning(X);

-Stacking Correction: Done: The correction function used for stacking ensemble performance evaluation. Evaluates performance and gives accuracy, phi, area under Receiver Operating Characteristic (ROC) curve, and accuracy as a mean of sensitivity and specificity. The input is the matrix of guesses from a classifier function (*ypred*) and the "answer key" to the testing data (*testing_label*). The required dimensions of the input are the matrix of classifier output and the "answer key" must be instances by one.

[phi,roc,auc_roc,accuracy,sensitivity,specificity,acc_sns,acc2,ppv,npv]=prototype_correctio
n_stacking(ypred,testing_label);

-Automated Run File: Done: A function that undertakes an automated battery of tests using 4 feature selection methods (PCA, PLS, ADEN, and ADENZ), with 3 types of classifier structures (single LDA classifier, AdaBoost, and stacking). Input values include the features (*features*), labels (*labels*), number of features (*pvalue*), number of AdaBoost weak learners (*itt*), and an identification number (*tag*). The results are stored in a separate '.mat' file, and successful completion will have the output value (*flag*) equal to the identification number (*tag*). Files would be named:

filename=['green.' num2str(tag) '.features.' num2str(pvalue) '.weaklearners.' num2str(itt) '.mat'];

[flag]=green(subs,features,labels,pvalue,itt,tag);

4. Feature Extraction Modules

Feature Extraction (FE) Modules**:** Feature Extraction modules generate feature-sets from preprocessed data being fed inside.

-Autoregressive Berg Sliding Window Function (BA): Done: Calculates power spectral density of input using a 40^{th} order Burg algorithm. Power ratios are also calculated. Generates 34 features. The output (*y*) is reduced in size from input (*segments*) and sampling frequency (*fs*).

[y]=feature_extraction_BA(segments,fs);

-Welch Sliding Window Function (PW): Done: Calculates power spectral density of input using the Welch method (averaging overlapping periodograms). Power ratios are also calculated. The same function is used for both AR coefficients and MFCCs. The input data (*segments*) and sampling frequency (*fs*) are needed. The output (*y*) is reduced in size from input (*segments*), and has 34 features.

[y]=feature_extraction_PW(segments,fs);

5. Feature Selection Modules

Feature Selection (FS) Modules: Feature Selection modules reduce the number of features and/or new features that have undergone various transforms.

-Principle Component Analysis (PCA): Done: PCA is an unsupervised method of FS. Redundant features are removed, generating a new feature set in the process. The input data must be in the format of features by instances (*y*). The testing data (*testing*) must be inserted in the format of features by instances. The number of features to keep is (*pvalue*). The outputs include the transformation matrices (*pcs*), variances (*var_exp* and *total_var_explained*), the transformed training data (*newf2*), and transformed testing data (*N2*). The output training and test matrices are likewise ordered in instances by (a reduced amount of) features.

[pcs,newf,var_exp,newf2,tot_var_explained,N2]=feature_selection_pca_alt(y,testing,pvalue)

-Average Distance between Event and Non-events (ADEN): Done: ADEN is a supervised method of FS, using distances between groups to rank features. The same dimensions of the matrices that go in come out of the process. The input must be in the format of instances by features (*training*). The target labels of the input data (*group*) must be in the format of instances by one. The testing data (*testing*) must be inserted in the format of instances by features. The total amount of features is also needed (*pvalue*). The outputs include the selected features (*w_aden*), max distance between classes (*a_aden*), the selected training data (*train_aden*), and selected testing data (*test_aden*). The output training and test matrices are likewise ordered in instances by features.

[w_aden,a_aden,training_aden,test_aden]=feature_selection_aden(training,group,testing,pvalue)

-Average Distance between Event and Non-events with Z score (ADENZ): Done: ADEN is a supervised method of FS, an early version of the ADEN module using a z-score transform to normalize the data. The same dimensions of the matrices that go in come out of the process. The input must be in the format of instances by features (*training*). The target labels of the input data (*group*) must be in the format of instances by one. The testing data (*testing*) must be inserted in the format of instances by features. The total amount of features is also needed (*pvalue*). The outputs include the selected features (*w_aden*), max distance between classes (*a_aden*), the selected training data (*train_aden*), and selected testing data (*test_aden*). The output training and test matrices are likewise ordered in instances by features.

[w_aden,a_aden,training_aden,test_aden]=feature_selection_adenz(training,group,testing,pvalue)

-Genetic Averaging between Events and Non-events (GADEN): Done: GADEN is a fusion of GA and ADEN, selecting features by implementing "fitness" requirements from a pool of top "ADEN" features. It is a supervised method of feature selection. The input must be in the format of instances by features (*training*). The target labels of the input data (*group*) must be in the format of instances by features. The testing data (*testing*) must be inserted in the format of instances by features. The total amount of features is also needed (*bottleneck*).The "pool" of top ADENs to select genes from (*limits*) is also necessary. In addition, the number of offspring for 3 generations is needed (*offspring*). The outputs include the selected features (*ga_ind*), max distance between classes (*maden*), the selected training data (*train_tng*), and selected testing data (*test_tng*). The output training and test matrices are likewise ordered in instances by features. The algorithm also performs ADEN, with the relevant outputs being selected training (*training_mad*), selected testing (*testing_mad*), selected features (*aden_ind*), and max distance (*maden*). In addition, the function can also perform "standard" GA by setting the "limits" to equal the number of features used.

[training_tng,test_tng,training_mad,test_mad,ga_ind,aden_ind,maden]=feature_selection_gaden(training,group,testing,limits,bottleneck,offspring)

-Projection to Latent Subspaces (PLS): Done: A supervised feature reduction technique, similar to PCA but incorporating class labels. The input data must be in the format of features by instances (*y*). The testing data (*testing*) must be inserted in the format of features by instances. The target labels of the input data (*group*) must be in the format of instances by one. The number of features to keep is (*pvalue*). The outputs include the transformed training data (*trainp*) and transformed testing data (*testp*).

[trainp,testp]=feature_selection_pls(y,group,testing,pvalue)

6. Pattern Recognition Modules

Pattern Recognition (PR) Modules: PR modules are used to classify data.

-Linear Discriminant Analysis (LDA): Done: An LDA classifier based on a modification of the code for the MATLAB default "classify" function. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance.

[ypred]=prototype_ldam_default_classify(test,training,training_label);

-Support Vector Machines (SVM): Gaussian Kernel: Done: Calls an SVM classifier using the Gaussian kernel via the SVM-KM toolbox. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance.

[ypred]=prototype_svm_default_classify(test,training,group);

-Support Vector Machines (SVM): Polynomial Kernel: Done: Calls an SVM classifier using the polynomial kernel via the SVM-KM toolbox. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output

vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance.

[ypred]=prototype_svm_poly_classify(test,training,training_label);

-Radial Basis Function (RBF): Done: Calls a Radial Basis Function Neural Net from the MATLAB code. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance.

[ypred]=prototype_rbf_default_classify(test,training,group);

-Stacking Linear Discriminant Analysis (LDA): Done: An LDA classifier based on a modification of the code for the MATLAB default "classify" function. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance normalized into binary labels. The other output vector (*outprobs*) has the dimensions of instances by one, and contains non-normalized outputs.

[ypred,outprobs]=stacking_ldam_default_classify(test,training,group);

-Support Vector Machines (SVM): Gaussian Kernel: Done: Calls an SVM classifier using the Gaussian kernel via the SVM-KM toolbox. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance normalized into binary labels. The other output vector (*outprobs*) has the dimensions of instances by one, and contains non-normalized outputs.

[ypred,outprobs]=stacking_svm_default_classify(testing,training,group);

-Support Vector Machines (SVM): Polynomial Kernel: Done: Calls an SVM classifier using the polynomial kernel via the SVM-KM toolbox. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance normalized into binary labels. The other output vector (*outprobs*) has the dimensions of instances by one, and contains non-normalized outputs.

[ypred,outprobs]=stacking_svm_poly_classify(testing,training,group);

-Stacking Radial Basis Function (RBF): Done: Calls a Radial Basis Function Neural Net from the MATLAB code. The inputs are the testing data, training data, and training targets. The dimension requirements for testing data (*test*), training data (*training*), and training target (*training_label*) matrices are instances by features. The output vector (*ypred*) has the dimensions of instances by one, and the contents of the output vector are class labels assigned to each instance normalized into binary labels. The other output vector (*outprobs*) has the dimensions of instances by one, and contains non-normalized outputs.

[ypred,outprobs]=stacking_rbf_default_classify(test,training,group);

7. Classifier Structure Modules

Classifier Structure (CS) Modules: CS modules are the concluding blocks of the system.

-Cross Validation: Cross-validation is a single classifier, based on the summed output of training a separate classifier for each subject. It is a concluding module for the system. Performs *x*-fold cross validation using feature selection and classification modules. The input required is the number of *x* subjects (*subs*), cell-based struct of features (*total_features*), number of features (*pvalue*), and cell-based struct of targets (*total_labels*). The output is a matrix of all mean metrics (*mean_measures*), as well as separate metrics averaged for each arrangement of the system, such as mean phi (*mean_phi*), phi calculated with another implementation (*mean_phiclassic*), mean accuracy (*mean_accuracy*), mean sensitivity (*mean_sensitivity*), and mean specificity (*mean_specificity*). The format for names is: (classifier module abbreviation)_(feature selection module abbreviation)_mval (e.g. lda_aden_mval)

[mean_measures,mean_phi,mean_phiclassic,mean_accuracy,mean_sensitivity,mean_specificity,mean_acc2,mean_ppv,mean_npv]=lda_pca_mval(subs,total_features,total_labels,pvalue);

-Bagging Cross Validation: Mixed cross-validation is an ensemble classifier, based on the summed output of training a separate classifier for randomized blocks of data taken from each subject and testing on a subject never seen before. It is a concluding module for the system. Performs *x*-fold cross validation using feature selection and classification modules. The input required is the number of *x* subjects (*subs*), cell-based struct of features (*total_features*), number of mixed blocks (*num_subs*), number of features (*pvalue*), and cell-based struct of targets (*total_labels*). The output is a matrix of all mean metrics (*mean_measures*), as well as separate metrics averaged for each arrangement of the system, such as mean phi (*mean_phi*), phi calculated with another implementation (*mean_phiclassic*), mean accuracy (*mean_accuracy*), mean sensitivity (*mean_sensitivity*), and mean specificity (*mean_specificity*). The format for names is: (classifier module abbreviation)_(feature selection module abbreviation)_mval (e.g. lda_aden_mval)

[mean_measures,mean_phi,mean_phiclassic,mean_accuracy,mean_sensitivity,mean_specificity,mean_acc2,mean_ppv,mean_npv]=lda_pca_nval(subs,total_features,total_labels,pvalue,num_subs);

-Mixed Cross Validation: Mixed cross-validation is a single classifier, based on the summed output of training a separate classifier for randomized blocks of data taken from each subject and testing on other blocks. It is a concluding module for the system. Performs *x*-fold cross validation using feature selection and classification modules. The input required is the number of *x* subjects (*subs*), cell-based struct of features (*total_features*), number of mixed blocks (*num_subs*), number of features (*pvalue*), and cell-based struct of targets (*total_labels*). The output is a matrix of all mean metrics (*mean_measures*), as well as separate metrics averaged for each arrangement of the system, such as mean phi (*mean_phi*), phi calculated with another implementation (*mean_phiclassic*), mean accuracy (*mean_accuracy*), mean sensitivity (*mean_sensitivity*), and mean specificity (*mean_specificity*). The format for names is: (classifier module abbreviation)_(feature selection module abbreviation)_mval (e.g. lda_aden_mval)

[mean_measures,mean_phi,mean_phiclassic,mean_accuracy,mean_sensitivity,mean_specificity,mean_acc2,mean_ppv,mean_npv]=lda_pca_pval(subs,total_features,total_labels,pvalue,num_subs);

-Stacking: Done: Stacking separates the training data into a training set and pseudo-testing set. The performance of the pseudo-testing data is used to train a meta-learner, which uses a validation subject to evaluate performance. A classification threshold is used to maximize accuracy and performance measures. Each subject is left out and used as the validation subject. Performance measures are averaged. The input required is the number of *x* subjects (*subs*), number of features (*pvalue*), cell-based struct of features (*total_features*), and cell-based struct of targets (*total_labels*). The output is a matrix of all mean metrics (*mean_measures*), as well as separate metrics averaged for each arrangement of the system, such as mean phi (*mean_phi*), phi calculated by another method (*mean_phiclassic*), mean accuracy (*mean_accuracy*), mean sensitivity (*mean_sensitivity*), and mean specificity (*mean_specificity*).

[mean_measures,mean_phi,mean_phiclassic,mean_accuracy,mean_sensitivity,mean_specificity,mean_acc2,mean_ppv,mean_npv]=lda_pca_mstack(subs,total_features,total_labels);

-Boosting: AdaBoost: Done: AdaBoost uses an ensemble of weak learners with weighted data points. AdaBoost code based on work by Dirk-Jan Kroon from the University of Twente was used as a basis due to its efficient runtime. The code was also modified for use in creating a boosting module. A classification threshold is used to maximize accuracy and performance measures. Each subject is left out and used as the validation subject. Performance measures are averaged. The input required is the number of *x* subjects (*subs*), cell-based struct of features (*total_features*), number of features (*pvalue*), number of weak learners (*itt*), and cell-based struct of targets (*total_labels*). The output is a matrix of all mean metrics (*mean_measures*), as well as separate metrics averaged for each arrangement of the system, such as mean phi (*mean_phi*), phi calculated by another method (*mean_phiclassic*), mean accuracy (*mean_accuracy*), mean sensitivity (*mean_sensitivity*), and mean specificity (*mean_specificity*). (NOTE: AdaBoost uses its own embedded weak linear classifier instead of the standard LDA, RBF, or SVM kernel.)

[mean_measures,mean_phi,mean_phiclassic,mean_accuracy,mean_sensitivity,mean_specificity,mean_acc2,mean_ppv,mean_npv]=lda_pca_adaboost(subs,features,labels,pvalue,itt);

8. Further Reading

To test that all functions work, run the file: “ictomidemo.m”

Useful references on classifier ensembles:

http://www.scholarpedia.org/article/Ensemble_learning

http://www.ece.stevens-tech.edu/~hhe/cpe695f09/lecturenotes/Lecture7

APPENDIX C: MICROSLEEP DATASET GUIDE

By John LaRocco

12 Nov 2014

This guide is intended to prepare unwary souls in dealing with the forsaken, aeon-old lore of prior studies. Microsleep and lapse detection has occurred at the Institute for over a decade, and we've amassed a substantial amount of rated data. I've primarily interacted with two particular EEG datasets, Study A (by Malik Peiris) and Study C (by Govinda Poudel).

Study A

Also Known As: Study 1, the Malik data, the Peiris dataset

Population Size: 8 (of an original 15)

Length: 2 sessions of 60-min each per subject

Sampling Frequency: 256 Hz for EEG, 64 for Gold Standard.

EEG Channels: 16 (originally 16 referential EEG channels, later converted to bipolar)

Referential Channel Order: Fp2, F4, C4, P4, O2, Fp1, F3, C3, P3, O1, F8, T4, T6, F7, T3, T5, Disconnected, Veog, Heog, Steering

Bipolar Channel Order: Fp1-F7, F7-T3, T3-T5, T5-O1, Fp2-F8, F8-T4, T4-T6, T6-O2, Fp1-F3, F3-C3, C3-P3, P3-O1, Fp2-F4, F4-C4, C4-P4, P4-O2.

Data Source Format: "EEG-Subject[subject number]-Session[session number].mat"

Label Source Format: "Lapses-Subject[subject number]-Session[session number].mat"

Notes: Source data is referential in format. Use above channels to convert to bipolar. For the "Gold Standard," it is sampled at 64 Hz. "flat" refers to tracking flat spots. "binVideo" refers to video microsleeps. "probBM" refers to either a flat spot or a video event. "defBM" refers to only events where both flat spot and video event are present. (0) for alert, (1) for event.

Label Guide: The current "Gold Standard" includes two types: "3" (setting video events and/or flat spots=1, all else=0) and "0" (definite microsleeps with a video event AND a flat spot).

Variants:

-Clean Bipolar (SABIS): A feature set with ICA, eye blinks, and other sections manually deleted. Used in prior literature.

Features: 544 (34 spectral features per channel)

Data Format: 8 cell array, with each of size 544 by number of observations.

Label Format: 8 cell array, with each holding a binary vector 1 by number of observations. (0) for alert, (1) for event.

Data Filename: *clean_s1eeg.mat* [variable called *clean_s1eeg*];

Label Filename: *clean_s1labels.mat* [variable called *clean_s1labels*];

-Raw Referential (SARUS): Referential EEG calculated based on the raw EEG, using 2-s sliding window. No observation is deleted or removed from it.

Features: 544 (34 spectral features per channel)

Data Format: 8 cell array, with each of size 544 by 7200 observations.

Label Format: 8 cell array, with each holding a binary vector 1 by 7200. (0) for alert, (1) for event.

Data Filename: *total_reeg.mat* [data is a variable called *total_reeg*];

Gold Standard 3 (Lapse) Label Filename: *total_labels_gs3.mat* [data is a variable called *total_labels*];

Gold Standard 0 (Lapse) Label Filename: *total_labels_gs0.mat* [data is a variable called *total_labels_ms*];

-Raw Bipolar (SABUS): Bipolar EEG calculated based on the raw EEG, using 2-s sliding window. No observation is deleted or removed from it.

Features: 544 (34 spectral features per channel)

Data Format: 8 cell array, with each of size 544 by 7200 observations.

Label Format: 8 cell array, with each holding a binary vector 1 by 7200. (0) for alert, (1) for event.

Data Filename: *total_beeg.mat* [variable called *total_beeg*];

Gold Standard 3 (Lapse) Label Filename: *total_labels_gs3.mat* [data is a variable called *total_labels*];

Gold Standard 0 (Lapse) Label Filename: *total_labels_gs0.mat* [data is a variable called *total_labels_ms*];

-Mixed Clean Bipolar (Peiris 2011 JNE Dataset): A feature set with ICA, eye blinks, and other sections manually deleted, only randomly recombined into 4 randomly sorted combinations of the original 8 subjects.

Features: 544 (34 spectral features per channel)

Data Format: 4 cell array, with each of size 544 by number of observations.

Label Format: 4 cell array, with each holding a binary vector 1 by number of observations. (0) for alert, (1) for event.

Data Filename: *s1_clean_data_gs3_mixed.mat* [variable called *total_features*];

Label Filename: *s1_clean_labels_gs3_mixed.mat* [variable called *total_labels*];

-Mixed Raw Referential: Referential EEG calculated based on the raw EEG, using 2-s sliding window. No observation is deleted or removed from it. Randomly recombined into 8 new combinations of original 8 subjects.

Features: 544 (34 spectral features per channel)

Data Format: 8 cell array, with each of size 544 by 7200 observations.

Label Format: 8 cell array, with each holding a binary vector 1 by 7200. (0) for alert, (1) for event.

Data Filename: *s1r_ref_data_gs3_mixed.mat* [data is a variable called *total_features*];

Label Filename: *s1r_ref_labels_gs3_mixed.mat* [data is a variable called *total_labels*];

-Mixed Raw Bipolar: Bipolar EEG calculated based on the raw EEG, using 2-s sliding window. No observation is deleted or removed from it. Randomly recombined into 8 new combinations of original 8 subjects.

Features: 544 (34 spectral features per channel)

Data Format: 8 cell array, with each of size 544 by 7200 observations.

Label Format: 8 cell array, with each holding a binary vector 1 by 7200. (0) for alert, (1) for event.

Data Filename: *s1r_bipolar_data_gs3_mixed.mat* [data is a variable called *total_features*];

Label Filename: *s1r_bipolar_labels_gs3_mixed.mat* [data is a variable called *total_labels*];

Study C

Also Known As: Study 2, first combined examination of EEG and fMRI

Population Size: 10 (of an original 20)

Valid Subject Numbers: 203, 207, 208, 210, 211, 213, 214, 216, 217, 220.

Sampling Frequency: 250 Hz for EEG.

Length: 1 session of 50-min each (save one subject with approximately 10 minutes deleted)

Channel Order: 'O2', 'O1', 'OZ', 'PZ', 'P4', 'CP4', 'P8', 'C4', 'TP8', 'T8', 'P7', 'P3', 'CP3', 'CZ', 'FC4', 'FT8', 'TP7', 'C3', 'FZ', 'F4', 'F8', 'T7', 'FT7', 'FC3', 'F3', 'FP2', 'F7', 'FP1', 'VEOG', 'EKG', 'PO5', 'PO3', 'P1', 'POZ', 'P2', 'PO4', 'CP2', 'P6', 'PO6', 'CP6', 'C6', 'PO8', 'PO7', 'P5', 'CP5', 'CP1', 'C1', 'C2', 'FC2', 'FC6', 'C5', 'FC1', 'F2', 'F6', 'FC5', 'F1', 'AF4', 'AF8', 'F5', 'AF7', 'AF3', 'FPZ'

EEG Channels: 64 (but many were deleted after ICA and preprocessing. Number of channels left depends on subject, but can range from 30-60)

Data Source Format: "[subject number] _50min_hpf_ica_icremoved.set"

Label Source Format: "[subject number] _50min_hpf_ica_icremoved.set"

Notes: The duration and type of events in the EEG is labeled in the ".set" files. In "EEG.event" for event types of the "BM" and "Sleep" types (as with others) is listed (in the number of samples) under "duration."

Label Guide: The 2 main "Gold Standards" for Study 1 are "1" (BMs and Sleep marked as 1, all else set to 0) and "2" (BMs, DIREs, and Sleep marked as 1, all else set to 0). "2" was discontinued due to not improving performance.

Variants:

-Raw Referential (SCRIS): Referential EEG calculated based on the raw EEG, using 2-s sliding window similar to Study A. No observation is deleted or removed from it, save for one subject with ~10 min of data deleted.

Features: 2040 (34 spectral features per channel, with a vector of 34 zeros added in for channels absent in some subjects)

Data Format: 10 cell array, with each of size 2040 by number of observations.

Label Format: 10 cell array, with each holding a binary vector 1 by number of observations. (0) for alert, (1) for event.

Data Filename: *total_s2eeg.mat* [variable called *total_s2eeg*];

Label Filename: *total_s2labels.mat* [variable called *total_s2labels*];

-Mixed Raw Referential: Referential EEG calculated based on the raw EEG, using 2-s sliding window similar to Study A. No observation is deleted or removed from it, save for one subject with ~10 min of data deleted. Randomly recombined into 10 new combinations of original 10 subjects.

Features: 2040 (34 spectral features per channel, with a vector of 34 zeros added in for channels absent in some subjects)

Data Format: 10 cell array, with each of size 2040 by number of observations.

Label Format: 10 cell array, with each holding a binary vector 1 by number of observations. (0) for alert, (1) for event.

Data Filename: *s2_ref_data_gs1_mixed.mat* [variable called *total_features*];

Label Filename: *s2_ref_labels_gs1_mixed.mat* [variable called *total_labels*];

APPENDIX D: ADEN CODE

By John LaRocco

```
%--------------------------------------------------------------------------
 % FEATURE_SELECTION_ADEN
 % This is copied and pasted from MATLAB.
 % Last updated: Oct 2014, J. LaRocco.

 % Details: Feature selection using ADEN to find frequency band with greatest distance.

 % Usage:
 %
[correct_ind,fittest,training_aden,test_aden]=feature_selection_aden(training,group,testing,li
mits)

 % Input:
 %  training: Matrix of features data from training subjects.
 %  testing: Matrix of feature data from testing subjects.
 %  group: Matrix of feature data from training subject labels (must be 0 or 1).
 %  pvalue: Number of features to keep.

 % Output:
 %  correct_ind: Selected feature index.
 %  fittest: Value of max differences.
 %  training_aden: Matrix of ADEN features.
 %  test_aden: Testing matrix after ADEN selection.


%--------------------------------------------------------------------------

limits=pvalue;
[train_instances,train_features]=size(training);
[test_instances,test_features]=size(testing);
training=squeeze(training);
data=training;
%sort data by binary group
R_0=find(group==0);
x0=data(R_0,:);
[x01,x02]=size(x0);
x0_mean=mean(x0);
X=x0_mean;
R_1=find(group==1);
x1=data(R_1,:);

[x11,x12]=size(x1);
x1_mean=mean(x1);
Y=x1_mean;
clear i;
```

```
sigs=[];
%calculate distance for each individual feature
for i=1:x12;
   g0=X(:,[i]);
   g1=Y(:,[i]);
vg0=var(data(R_0,[i]));
vg1=var(data(R_1,[i]));
sg0=sum(data(R_0,[i]));
sg1=sum(data(R_1,[i]));
mg0=mean(data(R_0,[i]));
mg1=mean(data(R_1,[i]));

s2g0=(1./(x01-1)).*(sg0-mg0).^(2);
s2g1=(1./(x11-1)).*(sg1-mg1).^(2);
xd0=x01-1;
xd1=x11-1;

%use Cohen's d to normalize

theop=(xd0*s2g0+xd1*s2g1)./(xd0+xd1);

val=abs(g0-g1)./sqrt(theop);

sigs(1,i)=val;


end

%remove any NaNs and replace with zeros so they're ignored
dist2=prototype_cleanup(sigs);

swe=sum(dist2);
yeh=sqrt(swe);
fittest=max(dist2);
%rank the distances in descending order
selection=sort(dist2,'descend');
%select the top ones
veiser=selection(1:limits);

correct_ind=[];
%retrieve the specific indices corresponding to each remaining feature
for khan=1:length(veiser);
   findvalue=veiser;

   corium=find(dist2==findvalue(khan));
   correct_ind(khan)=corium(1);
end
%rank the indices in ascending order
correct_ind=sort(correct_ind,'ascend');
```

```
%select the revelant features from the training and testing matrices
training_aden=data(1:train_instances,[correct_ind]);
test_aden=testing(1:test_instances,[correct_ind]);
test_aden=(prototype_cleanup(test_aden));
training_aden=(prototype_cleanup(training_aden));
```